Journal of WSCG

Cumulative Edition

No.1. & 2

Edited by

Vaclav Skala, University of West Bohemia, Czech Republic

Journal of WSCG

Cumulative Edition

No.1. & 2

Edited by

Vaclav Skala, University of West Bohemia, Czech Republic

Journal of WSCG Cumulative Edition No.1&2

Editor: Vaclav Skala
c/o University of West Bohemia, Univerzitni 8
CZ 306 14 Plzen
Czech Republic
skala@kiv.zcu.cz *http://www.VaclavSkala.eu*

Managing Editor: Vaclav Skala

Published and printed by:
Vaclav Skala – Union Agency
Na Mazinách 9
CZ 322 00 Plzen
Czech Republic http://www.UnionAgency.eu

Hardcopy: *ISBN 978-80-86943-64-0*

Journal of WSCG

Board of Reviewers

Meng, Weiliang (China)
Menotti, David (Brazil)
Mestre, Daniel,R. (France)
Meyer, Alexandre (France)
Michael, Despina (Cyprus)
Michels, Dominik (United States)
Monti, Marina (Italy)
Montrucchio, Bartolomeo (Italy)
Movania, Muhammad Mobeen (Pakistan)
Mukai, Tomohiko (Japan)
Mura, Claudio (Switzerland)
Nagai, Yukie (Japan)
Nah, Jae-Ho (Korea)
Nanni, Loris (Italy)
Nogueira, Keiller (Brazil)
Nurzynska, Karolina (Poland)
Nyul, Laszlo (Hungary)
Oliveira, Joao Fradinho (Portugal)
Oztimur Karadag, Ozge (Turkey)
Paiva, Jose Gustavo (Brazil)
Parsons, Paul (Canada)
Patane, Giuseppe (Italy)
Paul, Padma Polash (Canada)
Peethambaran, Jiju (India)
Penedo, Manuel (Spain)
Pina, Jose Luis (Spain)
Pobegailo, Alexander (Belarus)
Puig, Anna (Spain)
Ramos, Sebastian (Germany)
Rasool, Shahzad (Singapore)
Reddy, Pradyumna (India)
Rehfeld, Stephan (Germany)
Rind, Alexander (Austria)
Rupprecht, Christian (Germany)
Sadlo, Filip (Germany)
Saito, Shunsuke (United States)
Santagati, Cettina (Italy)
Saraiji, MHD Yamen (Japan)
Saru, Dhir (India)
Seipel, Stefan (Sweden)
Shesh, Amit (United States)
Shi, Xin (China)
Shimshoni, Ilan (Israel)
Schaefer, Gerald (United Kingdom)
Schmidt, Johanna (Austria)
Schultz, Thomas (Germany)
Schwarz, Michael (Switzerland)
Silva, Romuere (Brazil)
Silva, Samuel (Portugal)
Singh, Rajiv (India)
Solis, Ana Luisa (Mexico)
Soriano, Aurea (Brazil)
Souza e Silva, Lucas (Brazil)
Spiclin, Ziga (Slovenia)
Svoboda, Tomas (Czech Republic)
Tavares, Joao Manuel (Portugal)
Teixeira, Raoni (Brazil)
Theussl, Thomas (Saudi Arabia)
Tomas Sanahuja, Josep Maria (Mexico)
Torrens, Francisco (Spain)
Tytkowski, Krzysztof (Poland)
Umlauf, Georg (Germany)
Vasseur, Pascal (France)
Vazquez, David (Spain)
Veras, Rodrigo (Brazil)
Walczak, Krzysztof (Poland)
Wanat, Robert (United Kingdom)
Wang, Lili (China)
Wang, Ruizhe (United States)
Wang, Lisheng (China)
Wenger, Rephael (United States)
Wijewickrema, Sudanthi (Australia)
Wu, YuTing (Taiwan)
Wu, Jieting (United States)
Wuensche, Burkhard,C. (New Zealand)
Xiong, Ying (United States)
Xu, Tianchen (Hong Kong SAR)
Xu, Chang (China)
Yang, Shuang (China)
Yasmin, Shamima (United States)
Yoshizawa, Shin (Japan)
Yu, Hongfeng (United States)
Zheng, Jianping (United States)
Zhong, Li (China)

Journal of WSCG

Vol.23, No.1-2

Contents

A New Extension to Kernel Entropy Component Analysis for Image-based Authentication Systems

Sepehr Damavandinejadmonfared
Advanced Cyber Security Research Centre Dept. of Computing, Macquarie University Sydney, Australia
sepehr.damavandinejadmonfared@mq.edu.au

Vijay Varadharajan
vijay.varadharajan@mq.edu.au

Abstract

We introduce Feature Dependent Kernel Entropy Component Analysis (FDKECA) as a new extension to Kernel Entropy Component Analysis (KECA) for data transformation and dimensionality reduction in Image-based recognition systems such as face and finger vein recognition. FDKECA reveals structure related to a new mapping space, where the most optimized feature vectors are obtained and used for feature extraction and dimensionality reduction. Indeed, the proposed method uses a new space, which is feature wisely dependent and related to the input data space, to obtain significant PCA axes. We show that FDKECA produces strikingly different transformed data sets compared to KECA and PCA. Furthermore a new spectral clustering algorithm utilizing FDKECA is developed which has positive results compared to the previously used ones. More precisely, FDKECA clustering algorithm has both more time efficiency and higher accuracy rate than previously used methods. Finally, we compared our method with three well-known data transformation methods, namely Principal Component Analysis (PCA), Kernel Principal Component Analysis (KPCA), and Kernel Entropy Component Analysis (KECA) confirming that it outperforms all these direct competitors and as a result, it is revealed that FDKECA can be considered a useful alternative for PCA-based recognition algorithms

1. Introduction

Fundamentally data transformation is of importance in machine learning and pattern analysis. The goal is to, alternatively, represent the high-dimensional data into a typically lower dimensional form revealing the underlying format and structure of the data. There is a large amount of literature on data transformation algorithms and methods [1], [2]. A dominant research area in data transformation is known as the so-called spectral methods. In spectral methods, the bottom or top eigenvalues (spectrum) and their corresponding eigenvectors play the main role in feature extraction and dimensionality reduction especially in constructed data matrixes. Some recent spectral methods include locally linear embedding [3], isometric mapping [4], and maximum variance unfolding [5], to name a few. See the recent review papers [6], [7] for thorough reviews of several spectral methods for dimensionality reduction. One of the most powerful and well known methods in the mentioned area is Principal Component Analysis (PCA) [8] which has been used in numerous applications and algorithms in data classification and machine learning[9],[10]. However, PCA [11] is a linear method which may not be beneficial when there might exist non-linear patterns hidden in the data. Over the last few decades, there have been a number of advanced improvements on PCA trying to overcome the drawback of linearly transformation and make PCA influential when dealing with nonlinear data. A very well-known and influential method is Kernel Principal Component Analysis (KPCA) [12]. In Kernel PCA [13], PCA is performed in a kernel feature space which is non-linearly related to the input data. It is enabled using a positive semi-definite (psd) kernel function computing the inner products within the new space (kernel feature space). Therefore, constructing the so-called kernel matrix or the inner product matrix is vital. Then, using the top eigenvalues and their corresponding eigenvectors to perform metric MDS [14] will lead to kernel PCA data transformation method. Kernel PCA has extensive use in many different contexts. For instance, kernel PCA has been used in machine learning algorithms from data classification [15] to data denoising [16][17][18]. In [19], kernel PCA is introduced for face recognition systems. Kernel PCA also has been used in finger vein recognition algorithms [20]. In 2010 [21], R. Jenssen proposed Kernel Entropy Component Analysis KECA as a new extension to kernel PCA. Kernel

ECA is fundamentally different from other spectral methods in two ways explained as follows; (1): The data transformation reveals structure related to the Renyi entropy of the input space data set and (2): The method does not necessarily use the top eigenvalues and eigenvectors of the kernel matrix. Shekar in 2012 [22], implemented KECA on face data base claiming KECA outperforms KPCA for face recognition purpose. In this paper, we develop a new spectral data transformation method, which is fundamentally different from Kernel ECA in the following important way:

- In FDKECA the dimension of the feature space is dependent on the dimension of the input data, not the number of input data. It means no matter how many data to analyze, the dimension of kernel matrix (kernel feature space) is fixed.

The mentioned difference will make the following advantages FDKECA has over KECA:

- FDKECA is much less computationally expensive than KECA as the dimension of the feature space, where the optimal PCA axes are calculated, is just as high as the dimension of the input data. This leads to a much faster method than traditionally used KECA.

- FDKECA has lower error rate than KECA as the axes obtained from our proposed feature space will contribute to more efficiency and less dimension compared to KECA.

The reminder of this paper is organized as follows: Section 2 illustrates some examples of spectral data transformation methods of importance. Feature Dependent Kernel Entropy Component Analysis (FDKECA) is developed in Section 3. The image reconstruction method and eigenface analysis using FDKECA are developed in Section 4. A spectral clustering algorithm using FDKECA is developed in section 5. Experimental results are presented in section 6. Finally, section 7 concludes the paper.

2. Spectral Data Transformation

In this section, we explain the fundamentals of PCA, KPCA, and KECA with examples to comprehend spectral basic data transformation methods.

2.1. Principal Component Analysis (PCA)

A well-known spectral data transformation method is PCA. Let $X = [x_1, ..., x_n]$, where $x_t \in R^d$ and $t = [1, ..., N]$. As PCA is a linear method, the following transformation is sought assuming A is $[d \times d]$ such that $y_t \in R^d$ and $t = [1, ..., N] : Y_{pca} = AX$ where $Y_{pca} = [y_1, ..., y_n]$. Therefore, the sample correlation matrix of Y_{pca} equals to:

$$\frac{1}{N} Y_{pca} Y_{pca}^T = \frac{1}{N} AX(AX)^T = A \frac{1}{N} XX^T A^T \qquad (1)$$

The sample correlation matrix of X is $\frac{1}{N} XX^T$. Determining A such that $\frac{1}{N} Y_{pca} Y_{pca}^T = I$ is the goal. Considering eigen-decomposition, we will have $\frac{1}{N} XX^T = V \triangle V^T$,where $\triangle$ is a diagonal matrix of the eigenvalues $\delta_1, ..., \delta_n$ in descending order having the corresponding eigenvectors $v_1, ..., v_n$ as the columns of V. Substituting into (1), it can be clearly observed that $A = \triangle^{-1/2} V^T$ leads to the goal such that $Y_{pca} = \triangle^{-1/2} V^T X$.

Performing a dimensionality reduction from d to $l \leq d$ is often achieved by the projection of data onto a subspace spanned by the eigenvectors (principal axes) corresponding to the largest top l eigenvalues.

2.2. Kernel Principal Component Analysis (KPCA)

Scholkoft in 1998 proposed Kernel PCA which is a nonlinear version of PCA operating in a new feature space called kernel feature space. This space is non-linearly related to the input space. The nonlinear mapping function (kernel function) is given $\Phi : R^d \rightarrow F$ such that $x_t = \Phi(x_t), t = 1, ..., N$ and $\Phi = [\Phi(x_1), ..., \Phi(x_N)]$. After performing such mapping in input data, PCA if implemented in F, we need an expression for the projection of P_{U_i} of Φ onto a subspace of feature space principal axes, for example, top l principals. It can be given by a positive semi-definite kernel function or Mercer kernel [23] [24], $k_\sigma = R^d \times R^d \rightarrow R$ computes an inner product in the Hilbert space F:

$$k_\sigma(x_t, x'_t) = \langle \phi(x_t) \phi(x'_t) \rangle \qquad (2)$$

The $(N \times N)$ kernel matrix K is defined such that element (t, t') of the kernel matrix equals to $k_\sigma(x_t, x'_t)$. Therefore, $K = \Phi^T \Phi$ is the inner product matrix (Gram matrix) in F. Then, Eigen-decomposing the kernel matrix we have $K = EDE^T$ where E is the eigenvectors $e'_1, ..., e'_n$ column wise and their corresponding eigenvalues are in D - $\lambda_1, ..., \lambda_n$- . Williams in [25] discussed that the equivalence between PCA and KPCA holds in KPCA as well (kernel feature space). Hence, we have:

$$\Phi_{pca} = P_{U_i} \Phi = D_l^{1/2} E_l^T \qquad (3)$$

Where D_l is the top large l eigenvalues of K andE_l is their corresponding eigenvectors stored in columns. It means that projecting Φ onto spanned feature space (principal axes) is given by $P_{U_i} \Phi = \sqrt{\lambda_i} e_i^T$.

Considering the analogy in (3), $\Phi_{pca} = D_l^{1/2} E_l^T$ is the solution to the following optimization problem:

$$\Phi_{pca} = D_l'^{1/2} E_l'^T : \min_{\lambda_1', e_1', ..., \lambda_N', e_N'} 1^T (K - K_{pca})^2 .1. \quad (4)$$

Where $K_{pca} = \Phi_{pca}^T \Phi_{pca}$. Therefore, this procedure minimizes the norm of $K - K_{pca}$.

2.3. Kernel Entropy Component Analysis

Selection of the subspace where the data is projected onto is of importance in spectral methods, which is achieved based on the top or bottom eigenvectors in PCA and KPCA. In KECA, however, this stage is based on entropy estimate. Using entropy estimate, the data transformation from higher dimension to lower dimension is obtained by projecting the input data onto the axes, which contribute to the entropy estimate of input space. The procedure of entropy estimate in KECA is given as follows: The Renyi entropy function is defined by

$$H(P) = -\lg \int p^2(x) d(x) \quad (5)$$

Where p is probability density of the input data. Considering the monotonic nature of logarithmic function, (12) can be replaced by the following equation:

$$V(P) = \int p^2(x) d(x) \quad (6)$$

Estimating $V(p)$, (14) is given:

$$\hat{p}(x) = 1/N \sum_{x_t \epsilon S} k_\sigma(x, x_t) \quad (7)$$

$k(x, x_t)$ is the kernel centred matrix, then:
$\hat{V}(p) = 1/N \sum_{x_t \epsilon S}^{p} (x_t)$

$$1/N \sum_{x_t \epsilon S} 1/N \sum_{x_t \epsilon S} k_\sigma(x, x_t) = 1/N^2 1^T K 1 \quad (8)$$

where K is $k_\sigma(x, x_t)$ and 1 is an $(N \times 1)$ vector which contains all ones. The Renyi entropy estimating can be calculated for eigenvalues and eigenvectors of the Kernel matrix. It is defined as $K = EDE^T$, where D includes the eigenvectors, $\lambda_1, \lambda_2, ..., \lambda_N$, and E consists of eigenvalues, $\alpha_1, \alpha_2, ..., \alpha_N$. Finally, rewriting (15), we have:

$$(p) = 1/N^2 \sum_{1}^{N} (\sqrt{\lambda_i} \alpha_i^T 1)^2 \quad (9)$$

3. Feature Dependent Kernel Entropy Component Analysis (FDKECA)

In this section, we will go through PCA and KECA feature space in details and clarify our motivation to propose the new transformation method, and then FDKECA is introduced.

3.1. Defining the Feature Dependent Kernel Entropy transformation

Generally, in spectral data transformation methods, finding the most valuable principal axes (appropriate directions in the feature space) is of greatest importance. In PCA, for example, it is extracted linearly from the principal feature space. In KECA, however, these axes are extracted from kernel Entropy feature space as discussed in previous subsection. We define Feature Dependent Kernel Entropy Component Analysis as a k-dimensional data transformation method obtained by projecting input data onto a subspace spanned by principal kernel axis contributing to the feature dependent kernel Entropy space. Feature dependent kernel Entropy space is defined as follows:

Let $X = [x_1, ..., x_N]$, where $x_t \in R^d$ and $t = [1, ..., N]$. The nonlinear mapping function is given $\Phi : R^d \rightarrow F^d$ such that $x_t' = \Phi(x_t'), t = 1, ..., d$ where x_t' is an N dimensional vector including all of the t_{th} features from N input data. Explaining this, we have $\Phi = [\phi(x_1'), ..., \phi(x_d')]$. The use of a positive semi-definite kernel function or Mercer kernel computes an inner product in the new space F^d:

$$k_\sigma(x_t', x_{t'}') = \langle \phi(x_t') \phi(x_{t'}') \rangle \quad (10)$$

The $(N \times N)$ kernel matrix-we define that as K_{FDKECA} -is now defined such that element (t, t') of the kernel matrix is $k_\sigma(x_t', x_{t'}')$. Therefore, K_{FDKECA} is the Gram matrix or the inner product matrix in F^d. The next stage in FDKECA is to perform PCA on K_{FDKECA}. Note that the kernel matrix taken in FDKECA feature space (K_{FDKECA}) is totally different from that of KPCA.

Fig. 1. illustrates a brief flow diagram of reaching kernel Entropy feature space from scratch. As it is shown in Fig. 1, N input data are first mapped into kernel space by ϕ and then the Gram matrix (kernel matrix) is calculated using inner product. Note that the dimension of kernel matrix is equal to the number of input data- N. Eigen-decomposition is the next step where all eigenvalues and their corresponding eigenvectors are extracted and reordered in a descending manner from the greatest to the smallest value. After finding the kernel axes in this space, the kernel matrix, which represents the input data, is projected onto the kernel feature vectors (eigenvectors). The drawback to KECA is that the dimension of feature space and kernel matrix could become too high and as a result data transformation could be computationally expensive. In addition, finding the most

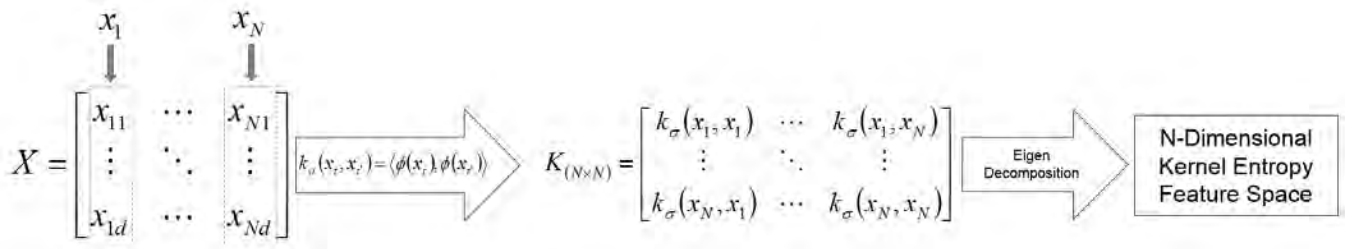


Figure 1. Flow diagram of reaching Kernel Entropy Feature Space

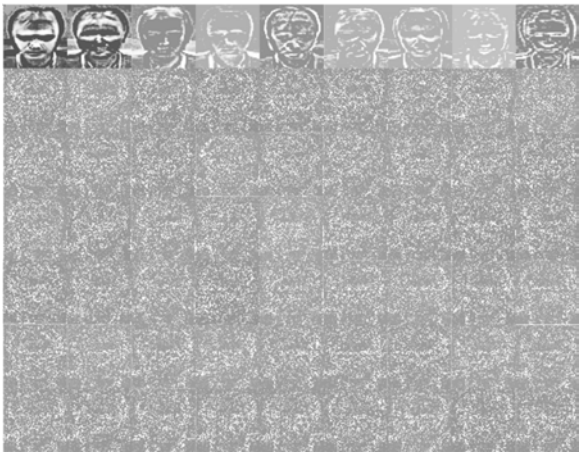


Figure 3. Top 63 eigenvectors obtained by PCA

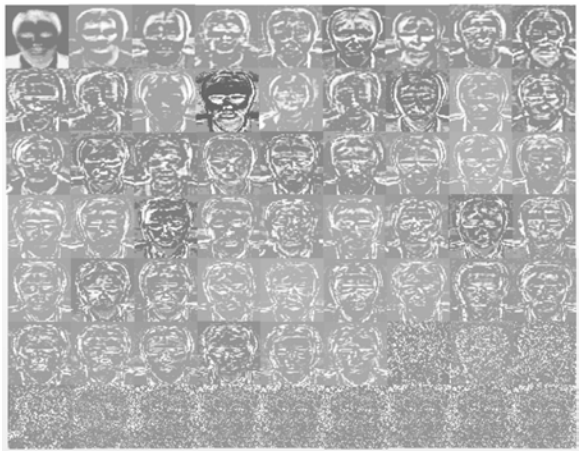


Figure 4. Top 63 eigenvectors obtained by FDKECA

optimized sub-space in kernel feature space could be challenging and sometimes inefficient.

Fig. 2. demonstrates FDKECA feature space where the input data is projected onto a subspace spanned by principal kernel entropy axes contributing to the feature dependent kernel entropy space. As it is illustrated in Fig.2, FDKECA considers all features having the same dimension from all input data in separate vectors first and then maps them into kernel space which is called FDKECA feature space. Then it computes the kernel matrix (Gram matrix) using inner products which is a d-dimensional space. Note that the input data has the dimension of d which means there is no growth of dimension while computing FDKECA feature space. Having d-dimensional FDKECA feature space, the eigenvectors and their corresponding eigenvalues are decomposed in this step using the estimation of entropy. The original input data is projected onto a sub-space of FDKECA feature vectors for the purpose of transformation and dimensionality reduction.

4. Eigenface Analysis on PCA and FDKECA

For more detailed comparison, we have performed PCA and FDKECA on the first individuals samples and visualized the first 63 feature vectors (eigenfaces) which are shown in Fig. 4 and 5.

In this analysis, we used 10 samples of the first subject of SCface database in PCA and FDKECA. In PCA, all samples were first converted into 1-D vectors. After calculating the mean vector (the mean image), the co-variance matrix is obtained and then, the Eigen-decomposition is performed on the co-variance matrix. The eigenvectors (PCA eigenfaces) were then reordered according to the greatness of their corresponding eigenvalues (in descending order). Fig. 4 shows the top 63 eigenvectors obtained by PCA. As it was expected, the top eigenvector carries the most information and the amount of information being carried by the feature vectors reduces as the eigenvector gets farther from the top one and closer to the bottom one. Another expectation is that only the first 9 or 10 top eigenvectors have some valuable information and the rest of the axes (eigenvectors) seem not to be useful as almost no related information can be seen in them. In terms of FDKECA, however, it is different.

In FDKECA, we used the polynomial kernel function with the degree of two. Firstly, all samples were converted into 1-D vectors. After calculating the mean vector (the mean image), all samples were mapped by the polynomial kernel function (as described in section III). Then, the Eigen-decomposition was performed on K_{FDKECA} to achieve the feature vectors and finally the axes were reordered based on entropy estimate. Fig. 5 illustrates the

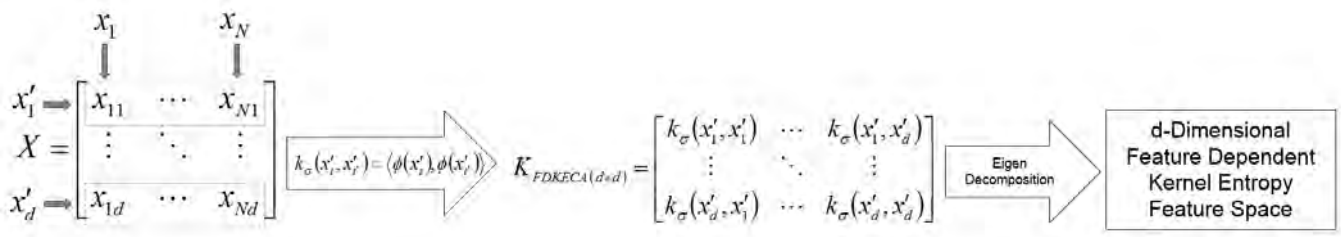


Figure 2. Flow diagram of reaching Feature Dependent Kernel Feature Space

top 63 eigenvectors obtained by FDKECA. Same as PCA, it was expected that the top eigenvector carries the most information and the amount of information drops as the number of the eigenvector gets closer to the bottom one. However, there is a considerable discrepancy between the shown eigenfaces obtained by PCA and FDKECA. In FDKECA, all eigenfaces carry relevant information except for the last 12 while in PCA only the first 9 or 10 ones have information related to the original face images. This analysis shows that FDKECA finds more informative and valuable feature vectors compared to PCA (as shown in Fig. 3 and4).

5. Spectral Clustring Algorithm Using FDKECA

In this section, a spectral clustering algorithm is developed using FDKECA transformation. The proposed algorithm, actually, is suitable for image classification which works in a supervised system as there are some samples to train the system and then using different samples, the system is tested. We first introduce the FDKECA clustering algorithm and then compare it with other algorithms such as PCA, KPCA and KECA in next section. As FDKECA can be considered as an extension to 1-D PCA, in our clustering algorithm all samples are converted into vectors. The goal is to propose a clustering system which not only is fast enough (not as computationally expensive as KECA), but also outperforms PCA, KPCA and KECA in terms of clustering image samples. Such an algorithm can be used in recognition systems like face, finger print, finger vein, palm vein etc.

Fig.5 indicates the flow diagram of the proposed clustering algorithm for image classification. We believe this algorithm can be applied in image-based recognition systems such as face and finger vein recognition. Moreover, this algorithm is much faster than normal KECA as its dimension of feature vector is fixed and it does not become too computationally expensive when analyzing a huge number of data. In addition to having a high speed, this algorithm is believed to be more appropriate than PCA, KPCA and KECA as it was shown in previous section. We have conducted different experiments on two different databases to have a complete analysis on the proposed algorithm. Next section gives experimental results on face and finger vein database.

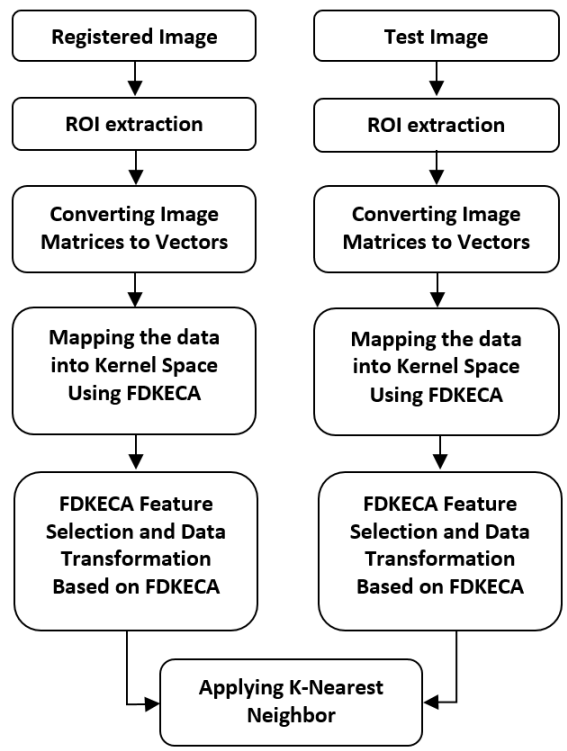


Figure 5. Flow diagram of the proposed clustring algorithm using FDKECA

6. Experimental Results

In this section, the performance of FDKECA is evaluated and compared with PCA, KPCA, and Kernel Entropy Component Analysis (KECA) on two different databases- finger vein and face. The experiments are conducted on Surveillance Camera Face Database (SCface database) and Finger vein database which are explained in two experimental setups in the following part of this section.

6.1. Experimental setup-1

The first part of the experiments is on finger vein database. The finger vein samples are collected using our own designed scanner. We will not go through the detailed discussion on how the data is collected and prepared as it might not be totally relevant to this work, See [27] for more information on the database.

10 samples were used from each of 200 individuals which results in a finger vein database consisting of 2000 samples. Region of Interest is detected and extracted from each sample automatically. Fig. 6 shows an original and cropped sample from the database. Two independent experiments have been conducted on this database. Firstly, the performance of FDKECA is compared with PCA, KPCA, and KECA where 5 randomly selected samples were used to train the algorithm and the remaining 5 to test. Then we used leave-one-out strategy to have a better comparison. Gaussian kernel is used in FDKECA, KPCA, and KECA algorithms in this stage. As in PCA-based image analysis the size of the samples is of importance, all finger vein samples have been normalized to the size of (10×20) to have a balance between speed and efficiency. In one-dimensional PCA-based algorithms, the first step is to convert the data from matrices into vectors which leads into vectors with the dimension of (1×200). It means there could be 200 different implementations of FDKECA on the data using 200 different feature vectors to project the data onto. However, it is totally different in KPCA and KECA as it is dependent on the number of input data being transferred into kernel space. For the sake of comparison, the first 200 kernel feature vectors were used in our implementations. In each single experiment, the implementation is repeated 200 times and the maximum accuracies and their corresponding dimension of feature vector are gathered and shown in Table 1. As it is observed from this table, KPCA and KECA achieve their maximum accuracy in a much higher dimension of feature vector in comparison with PCA. It is because feature space in KPCA and KECA is very high dimensional. more precisely, if 9 image from each category is used to train, it leads to a total number of 1800 train samples as there are 200 individuals. Having 1800 input samples in KPCA and/or KECA will result in a feature space with the dimension of (1800×1800) , while in PCA the dimension is fixed and equal to 200 in this experiment. The FDKECA, however, results in having the highest accuracy rate while its dimension of feature vector is almost as high as PCA, which means this method is not computationally as expensive as KPCA and KECA. Moreover, there is a dramatic gap between FDKECA and KECA which is more than 10 percent in the first experiment.

Table 1. Comparison of FDKECA with Other Methods Using the finger vein Database

Strategy	Method	Max Acc %	Dimension
5 for training	$KPCA$	85.9	200
	$KECA$	86	175
	PCA	95.3	68
	$FDKECA$	97.2	46
Leave-one-out	$KPCA$	92.5	173
	$KECA$	93.5	86
	PCA	98.5	35
	$FDKECA$	99.4	85

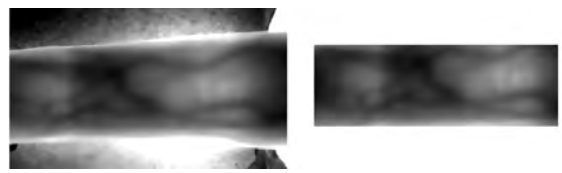


Figure 6. Original and ROI extracted finger vein sample

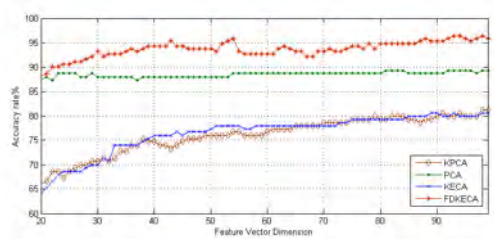


Figure 7. SCface classification using images of 4 cameras for training and 1 to test

6.2. Experimental setup-2

In the second experimental setup, we chose SCface database which is already explained in section 4. There are five different cameras located in three different distances from the individuals to collect the face data. In this part, we conducted the experiment using the images of 4 randomly selected cameras for training and the remaining 1 camera for testing. For each algorithm, the experience was repeated 100 times using the first 100 different eigenvectors to project the data onto and the results were gathered and visualized in Fig. 7. It is observed that Like the previous setup, FDKECA outperforms PCA, KPCA, and KECA in all experiments. As Fig.7 indicates, FDKECA reaches the highest accuracy of almost %98 while PCA, KPCA, and KECA get the accuracy of %89, %79 and %81 respectively.

7. Conclusion

We introduced a new data transformation method in this research work for dimensionality reduction in image-based recognition systems. Feature Dependent Kernel Entropy

Component Analysis (FDKECA) is an extension to both 1D-PCA and 1D-KECA. In FDKECA, all data is mapped into kernel space feature-wisely which results in having a constant dimension of data as well as being able to extract more valuable feature vectors in FDKECA feature space. Eigenface analysis showed that the feature vectors in FDKECA feature space are more informative than PCA. To examine FDKECA in practical clustering and classification methods and to be able to have a complete comparison with PCA, KPCA, and KECA, we proposed a clustering algorithm using FDKECA which was examined in two different areas- face recognition and finger vein recognition. Experimental results showed that FDKECA outperforms PCA, KPCA, and KECA which shows the reliability of FDKECA to be applied in image classification and recognition systems.

References

[1] R.O. Duda, P. E. Hart, and D.G. Stork *Pattern Classification.*, John Wiley and Sons, 2001.

[2] S. Theodoridis and K. Koutroumbas, *Pattern Recognition.* Academic Press, 1999.

[3] S. Roweis and L. Saul, *Nonlinear Dimensionality Reduction by Locally Linear Embedding,* Science, vol. 290, pp. 2323-2326, 2000.

[4] J. Tenenbaum, V. de Silva, and J.C. Langford, *A Global Geometric Framework for Nonlinear Dimensionality Reduction,* Science, vol. 290, pp. 2319-2323, 2000.

[5] K.Q. Weinberger and L.K. Saul, *Unsupervised Learning of Image Manifolds by Semidefinite Programming,* Intl J. Computer Vision, vol. 70, no. 1, pp. 77-90, 2006.

[6] L.K. Saul, K.Q. Weinberger, J.H. Ham, F. Sha, and D.D. Lee, *Spectral Methods for Dimensionality Reduction,* Semisupervised Learning, O. Chapelle, B. Scholkopf, and A. Zien, eds., chapter 1, MIT Press, 2005.

[7] C.J.C. Burges, *Geometric Methods for Feature Extraction and Dimensional Reduction,* Data Mining and Knowledge Discovery Handbook: A Complete Guide for Researchers and Practitioners, O. Maimon and L. Rokach, eds., chapter 4, Kluwer Academic Publishers, 2005.

[8] Jolliffe, I. 2005. *Principal Component Analysis. Encyclopedia of Statistics in Behavioral Science.*

[9] L. Sirovich and M. Kirby, *Low-Dimensional Procedure for Characterization of Human Faces,* J. Optical Soc. Am., vol. 4, pp. 519-524, 1987.

[10] M. Kirby and L. Sirovich, *Application of the KL Procedure for the Characterization of Human Faces,* IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 12, no. 1, pp. 103-108, Jan. 1990.

[11] I.T. Jolliffe, *Principal Component Analysis. Springer Verlag*, 1986.

[12] B. Scholkopf, A.J. Smola, and K.-R. Mu ller, *Nonlinear Component Analysis as a Kernel Eigenvalue Problem,* Neural Computation, vol. 10, pp. 1299-1319, 1998.

[13] Schlkopf, Bernhard, Alexander Smola, and Klaus-Robert Mller. *"Kernel principal component analysis."* Artificial Neural NetworksICANN'97. Springer Berlin Heidelberg, 1997. 583-588.

[14] H. Hotelling, *Analysis of a Complex of Statistical Variables into Principal Components,* J. Educational Psychology, vol. 24, pp. 417- 441, 1933.

[15] M.L. Braun, J.M. Buhmann, and K.-R. Mu ller, *On Relevant Dimensions in Kernel Feature Spaces,* J. Machine Learning Research, vol. 9, pp. 1875-1908, 2008.

[16] J.T. Kwok and I.W. Tsang, *The Pre Image Problem in Kernel Methods,* IEEE Trans. Neural Networks, vol. 15, no. 6, pp. 1517- 1525, 2004.

[17] S. Mika, B. Scholkopf, A. Smola, K.R. Mu ller, M. Scholz, and G. Ratsch, *Kernel PCA and Denoising in Feature Space,* Advances in Neural Information Processing Systems, 11, pp. 536-542, MIT Press, 1999.

[18] B. Scholkopf, S. Mika, C.J.C. Burges, P. Knirsch, K.-R. Mu ller, G. Ratsch, and A.J. Smola, *Input Space versus Feature Space in Kernel-Based Methods,* IEEE Trans. Neural Networks, vol. 10, no. 5, pp. 1299-1319, 1999.

[19] Kim, Kwang In, Keechul Jung, and Hang Joon Kim. *"Face recognition using kernel principal component analysis."* Signal Processing Letters, IEEE 9.2 (2002): 40-42.

[20] Damavandinejadmonfared, Sepehr, et al. *"Finger Vein Recognition using PCA-based Methods."* World Academy of Science, Engineering and Technology 6.6 (2012): 1079-1081.

[21] Jenssen, Robert. *"Kernel entropy component analysis."* Pattern Analysis and Machine Intelligence, IEEE Transactions on 32.5 (2010): 847-860.

[22] Shekar, B. H., et al. *"Face recognition using kernel entropy component analysis."* Neurocomputing 74.6 (2011): 1053-1057.

[23] J. Mercer, *Functions of Positive and Negative Type and Their Connection with the Theory of Integral Equations,* Philosophical Trans. Royal Soc. London, vol. A, pp. 415-446, 1909.

[24] K.R. Mu ller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, *An Introduction to Kernel-Based Learning Algorithms,* IEEE Trans. Neural Networks, vol. 12, no. 2, pp. 181-201, Mar. 2001.

[25] C.K.I. Williams, *On a Connection between Kernel PCA and Metric Multidimensional Scaling,* Machine Learning, vol. 46, pp. 11-19, 2002.

[26] Grgic, Mislav, Kresimir Delac, and Sonja Grgic. *"SCfacesurveillance cameras face database."* Multimedia tools and applications 51.3 (2011): 863-879.

[27] S.Damavandinejadmonfared, and V.Varadharajan. *Finger Vein Recognition in Row and Column Directions Using Two Dimensional Kernel Principal Component Analysis* Proceedings of the 2014 International Conference on Image Processing, Computer Vision Pattern Recognition, IPCV14, July 2014, USA.

Cloth Simulation Using Soft Constraints

Mihai Frâncu
Politehnica University
Bucharest, Romania
mihai.francu@cs.pub.ro

Florica Moldoveanu
Politehnica University
Bucharest, Romania
florica.moldoveanu@cs.pub.ro

Abstract

This paper describes a new way of using projective methods for simulating the constrained dynamics of deformable surfaces. We show that the often used implicit integration method for discretized elastic systems is equivalent to the projection of regularized constraints. We use this knowledge to derive a Nonlinear Conjugate Gradient implicit solver and a new projection scheme based on energy preserving integration. We also show a novel way of adding damping to position based dynamics and a different view on iterative solvers. In the end we apply these fresh insights to cloth simulation and develop a constraint based finite element method capable of accurately modeling thin elastic materials.

Keywords

Implicit integration, constraints, projection, PBD, damping, iterative solver, cloth, FEM

1 INTRODUCTION

For decades now the preferred method of simulating elastic systems in computer graphics has been the implicit integration of the equations of motion. The method is very attractive due to its unconditional stability and large time steps. It has been widely used for simulating cloth and finite difference or finite element soft bodies in general. Constraint based methods on the other hand have not received that much attention, with the exception of rigid body dynamics. Only recently there has been an increase in the number of papers on the subject in relation to soft bodies and we believe there is room for improvement. Many regard the method as being an inaccurate approximation of natural phenomena which are better described by elasticity theory. In this paper we aim to reconcile the two methods and show that they are two faces of the same problem; this can prove useful for the further development of both approaches.

1.1 Related work

Constraint based methods have appeared originally in their acceleration based formulation for rigid body dynamics [Bar94]. Later on, velocity or impulse based methods gained more popularity [AH04, Erl07]. Position based methods are actually a nonlinear version of velocity based ones, in the sense that they can still be expressed as velocity filters, but constraints are enforced at positional level [ST96]. Such a method was made popular in games by [Jak01] and was later refined and extended by [MHHR07] under the name Position Based Dynamics (PBD). Part of the inspiration for this method came from molecular dynamics where methods like SHAKE or RATTLE are widely used [BKLS95]. A more detailed study for the application to cloth simulation in computer graphics was done in [Gol10]. Here the method of *fast projection* is developed based on an implicit treatment of constraint directions [HCJ*05] and a better energy preserving integrator is also derived. A similar method was used to develop the unified Nucleus solver in Autodesk Maya [Sta09]. Position based methods rely on *projection* for solving differential algebraic equations (DAE), which is ultimately an optimization problem [HLW06]. Another part of inspiration came from *strain limiting* techniques used in elastic cloth simulation [Pro96, BFA02].

Constraint based methods are often criticized for the fact they simulate only nearly inextensible materials and are prone to *locking*. In order to address this [EB08] use fast projection in conjunction with a BDF-2 integrator on a conforming triangular mesh. They also give a brief proof for fast projection being the limit of infinitely stiff elastic forces. Other authors prefer to use quad-predominant meshes or diamond subdivision [Gol10].

Constraint *regularization* was employed mainly in [Lac07] for making rigid dynamics with contact and friction more tractable numerically. We take the name *soft constraints* from [Cat10] where an older idea is used: regularization under the mask of Constraint

Force Mixing (CFM) [Smi06]. Recently constraint regularization has been used for particle based fluid simulation [MM13]. Another application was intended for the simulation of deformable elastic models using a constraint based formulation of the linear Finite Element Method (FEM) [SLM06]. Similar position based approaches can be found in [BML*14] and [BKCW14]. The FEM constraint approach is similar in philosophy with *continuum strain limiting* [TPS09].

The implicit integration of the equations of motion has become pervasive for cloth since the seminal work of [BW98]. The method was also applied for FEM simulation [MSJT08]. Its main attraction is its unconditional stability for very stiff equations and large time steps. By implicit integration we usually mean the Implicit Euler (IE) method, but other implicit integrators were also employed, like BDF-2 [CK02], Implicit Midpoint [OAW06] or Newmark [SSB13]. These integration methods offer better energy conservation and more responsive simulation, in contrast to IE which artificially dampens out high frequency details in exchange for stability. Other variations include approximations made to the force Jacobian [HET01] or an implicit-explicit (IMEX) approach [EEH00, BMF03]. Most approaches however use only one Newton solver iteration. More recently a new view on IE as an optimization problem was presented in [LBOK13].

A special class of integrators labeled *variational* can be deduced directly from the discretization of the Euler-Lagrange equations of motion [SD06]. They are also symplectic integrators, i.e. they preserve area in phase space, which also means they are closer to preserving energy and momenta [HLW06]. Many of them are explicit methods (e.g. Symplectic Euler, Verlet, Leapfrog) so care must be taken to the time step size. Variational implicit methods like Implicit Midpoint or Newmark are more stable and can be converted to projection schemes (e.g. through our constraint space transformation). This is why we used them as inspiration for our energy conservation strategy.

A new alternative that is totally different from implicit integration of elastic systems or our approach is *exponential integration* [MSW14] which relies on evaluating trigonometric matrices (in terms of exponential functions).

1.2 Contributions

We present in this paper a constraint based simulator that is able to reproduce fully the elastic properties of cloth. We base our results on the fact that constraint projection methods are in fact equivalent to implicit integration of stiff springs (Section 2). Our approach is not entirely new as it is based on the idea of constraint regularization [Lac07]. We chose to use a PBD method instead as it corresponds to the nonlinear case [ST96] and it handles fast deforming bodies more robustly. Catto [Cat10] uses Baumgarte stabilization (ERP) and CFM and relates them to the stiffness and damping of an implicit harmonic oscillator. We give a more general and accurate correspondence to elastic parameters.

In Section 3 we derive a simple implicit integration solver based on the Variational Implicit Euler approach in [LBOK13] and the Nonlinear Conjugate Gradient method which is very similar to PBD. From it we obtain the equations of regularized constraint projection (Section 4). Using this transformation we derive a new projection method with better energy conservation (Section 5). In Section 6 we present a novel and effective way of adding more damping to PBD. Section 7 shows how *relaxation* can be used for block solving and regularization and how to transform the Conjugate Gradient method into Conjugate Residuals. In Section 8 we present constrained mass-spring systems for cloth and how to prevent locking. In Section 9 we present a nonlinear Saint Vennant-Kirchoff (StVK) elasticity model implemented through soft constraints. We take the area and planar strain constraint from [SLM06] and derive a method that takes into account the discretization and elastic properties of cloth. The closest approach to our method is [BKCW14] but they use energy as a constraint instead of a minimization objective. [BML*14] is using the same energy objective as us but their numerical method is different.

2 OPTIMIZATION EQUIVALENCE

In this section we would like to show that implicitly integrating stiff elastic forces is no different than using a constraint based formulation with regularization. The most general way to show this is by employing an optimization formulation for both methods. Let us start with Implicit Euler:

$$\mathbf{M}\Delta\mathbf{v} = h\mathbf{f}(\mathbf{x}_0 + \Delta\mathbf{x}), \tag{1}$$

$$\Delta\mathbf{x} = h\Delta\mathbf{v} \tag{2}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{x}$ are positions, $\mathbf{v}$ are velocities, h is the time step, $\mathbf{x}_0 = \mathbf{x}^{(n)} + h\mathbf{v}^{(n)}$ and $\mathbf{f}(\mathbf{x})$ are conservative forces, i.e. $\mathbf{f}(\mathbf{x}) = -\nabla_{\mathbf{x}} U(\mathbf{x})$. We can reformulate (1) as:

$$\tfrac{1}{h^2}\mathbf{M}\Delta\mathbf{x} = -\nabla_{\mathbf{x}} U(\mathbf{x}_0 + \Delta\mathbf{x}), \tag{3}$$

This is a nonlinear equation which is typically solved using the Newton method, but it can also be regarded as the optimality condition of an optimization problem:

$$\text{minimize } \tfrac{1}{2h^2}\Delta\mathbf{x}^T\mathbf{M}\Delta\mathbf{x} + U(\mathbf{x}_0 + \Delta\mathbf{x}). \tag{4}$$

It is shown in [GHF*07] that a similar formulation can be used for a *projection* method using Lagrange multipliers and implicit constraint directions:

$$\text{min. } \tfrac{1}{2h^2}\Delta\mathbf{x}^T\mathbf{M}\Delta\mathbf{x} - \lambda^T\mathbf{c}(\mathbf{x}_0 + \Delta\mathbf{x}) + U_{ext}(\mathbf{x}^{(n)}), \tag{5}$$

where $\mathbf{c}(\mathbf{x})$ are the constraint functions that need to be zero. Note that we modified the formulation so that the external forces potential is included in the objective function. Note that the external forces are treated explicitly, i.e. using the position at the beginning of the frame, which is also the case for (4).

The potential term $U_c = \lambda^T \mathbf{c}(\mathbf{x})$ gives us the internal constraint forces. If we take the gradient of this constraint potential we get the *principle of virtual work*:

$$\mathbf{f}_c = \nabla_{\mathbf{x}} \mathbf{c}(\mathbf{x})\lambda = \mathbf{J}^T \lambda. \tag{6}$$

If we express the total potential in general as $U = U_{int} + U_{ext}$, then so far the objectives in (4) and (5) are the same in the first (inertial) term and U_{ext}. For U_{int} we have an expression in the second case, but we have not yet specified one for the implicit integration. And we are not forced to provide one, but in reality, following the approach in [BW98], this internal potential is usually made up of quadratic elastic potentials with the purpose of enforcing certain constraints [Lan70]:

$$U_e(\mathbf{x}) = \tfrac{k}{2} \|\mathbf{c}(\mathbf{x})\|^2. \tag{7}$$

More generally we can replace stiffness k by a matrix:

$$U_e(\mathbf{x}) = \tfrac{1}{2} \mathbf{c}(\mathbf{x})^T \mathbf{E} \mathbf{c}(\mathbf{x}),$$

which is extremely useful when dealing with different stiffnesses in a mass-spring system or continuum based constitutive laws.

The potential energy in (7) gives forces of the form:

$$\mathbf{f}_e(\mathbf{x}) = -k\mathbf{J}^T \mathbf{c}(\mathbf{x}). \tag{8}$$

By comparing (6) and (8) we can see that they act in the same direction and by requiring that they have the same magnitude we obtain the regularization condition:

$$\mathbf{c}(\mathbf{x}) + \varepsilon\lambda = 0, \tag{9}$$

where $\varepsilon = 1/k$. We call this a *soft constraint* and by enforcing it we basically set the internal potential energy in (5) to be the same as in (4), thus making the two problems equivalent. It is clear now that when stiffness k goes to infinity ($\varepsilon \to 0$) implicit integration of springs becomes the constrained dynamics problem in (5). In the general case ε gets replaced by $\mathbf{E}^{-1}$.

In conclusion, not only is constraint based dynamics a limit case of implicit integration, but it can be made equivalent by replacing the strict constraint condition with a "softer" one. This permits us to solve a mass-spring system or any other discretized elastic system by casting the problem into the following form:

$$\begin{aligned} \mathbf{M}\Delta\mathbf{x} &= h^2 \left(\mathbf{J}^T \lambda - \nabla_{\mathbf{x}} U_{ext}(\mathbf{x}^{(n)}) \right), \\ 0 &= \mathbf{c}(\mathbf{x}_0 + \Delta\mathbf{x}) + \varepsilon\lambda \end{aligned}$$

This equivalence opens up a whole range of opportunities, especially for bringing results from implicit integration into the world of constraint based simulation. This was not considered possible in the past, as projection methods were regarded as an approximation of true elasticity based ones [Lac07, LBOK13].

3 NONLINEAR CONJUGATE GRADIENT SOLVER

The most important analogy we make in this paper is that between Implicit Euler integration and PBD. PBD starts from a candidate position (that includes the effect of external forces) and then runs an iterative process that does not involve the second derivative of the constraint function. This process is actually a minimization algorithm based on *sequential quadratic programming* (SQP) [WN99] that involves solving a linear system at every iteration, called *fast projection* [GHF*07]. This process can be further optimized by employing an inexact one step SOR-Newton scheme [Jak01, MHHR07] that reduces the cost of each iteration by running only one relaxation step.

The same logic can be applied to the Implicit Euler method expressed as the quadratic minimization problem in (4). If we choose the initial guess state to be one that incorporates the external forces, i.e. positions and velocities after an unconstrained step, we arrive at an approach similar to fast projection. The only difference is that the former works in configuration space, while the latter works in constraint space.

If we consider the initial candidate state consisting of $\tilde{\mathbf{v}} = \mathbf{v}^{(n)} + h\mathbf{f}_{ext}$ and $\tilde{\mathbf{x}} = \mathbf{x}^{(n)} + h\tilde{\mathbf{v}}$ we can rewrite (1) using a first order Taylor expansion around $\tilde{\mathbf{x}}$:

$$\mathbf{M}\delta\mathbf{v} = h\left(\mathbf{f}(\tilde{\mathbf{x}}) + \mathbf{K}\delta\mathbf{x}\right), \tag{10}$$

where $\mathbf{K} = \nabla_{\mathbf{x}}\mathbf{f} = -\frac{\partial^2 U}{\partial \mathbf{x}^2}$ is the *tangential stiffness matrix* and $\delta\mathbf{x} = h\delta\mathbf{v}$. Most authors choose to solve the implicit integration problem using only one Newton step, meaning we only need to solve one single linear system: $\mathbf{S}\delta\mathbf{v} = \mathbf{t}$. This works well in practice, but only if $\mathbf{K}$ contains second derivatives of the constraint function. This is because these terms contain information about the change of the constraint direction, so without them we need an iterative algorithm that keeps updating the constraint gradient. By dropping the second derivative term from $\mathbf{K}$ (see [BW98]) we get:

$$\mathbf{K} = -k\mathbf{J}^T\mathbf{J}. \tag{11}$$

This is equivalent to linearizing the force in (8) as in [EB08]. Using this formula at every Newton iteration we get the series of linear systems we need to solve:

$$(\mathbf{M} + h^2 k \mathbf{J}_i^T \mathbf{J}_i)\delta\mathbf{v}_{i+1} = h\mathbf{f}(\mathbf{x}_i), \tag{12}$$

where $\mathbf{J}_i^T = \nabla_{\mathbf{x}}\mathbf{c}(\mathbf{x}_i)$ and $\mathbf{x}_{i+1} = \mathbf{x}_i + h\delta\mathbf{v}_{i+1}$.

Uncostrained step to $\tilde{\mathbf{x}}, \tilde{\mathbf{v}}$
Compute Jacobian $\mathbf{J}$ and forces $\mathbf{f}$ using (8)
Compute residual $\mathbf{r} = \mathbf{d} = h\mathbf{f}$ and its square $\delta = \mathbf{r}^2$
for iter = 1:maxIter **do**
  Compute $\mathbf{q} = \mathbf{S}\mathbf{d} = (\mathbf{M} + h^2 k \mathbf{J}^T \mathbf{J})\mathbf{d}$
  Compute impulse $\mathbf{p} = \alpha \mathbf{d}$, where $\alpha = \delta / \mathbf{q}^T \mathbf{d}$
  Integrate: $\mathbf{v} \leftarrow \mathbf{v} + \mathbf{p}$, $\mathbf{x} \leftarrow \mathbf{x} + h\mathbf{p}$
  Recompute Jacobian $\mathbf{J}$ and forces $\mathbf{f}$
  Compute residual $\mathbf{r} = h\mathbf{f}$ and its square $\delta' = \mathbf{r}^2$
  Compute $\beta = \delta' / \delta$ and then $\delta' \leftarrow \delta$
  Compute new search direction $\mathbf{d} = \mathbf{r} + \beta \mathbf{d}$

Algorithm 1: NCG implicit solver

Nonlinear Conjugate Gradient (NCG) [She94] is a natural solution for solving the above problem, given its linear version is very popular for solving the one Newton step approach. The only changes we need to make to linear CG is to replace the system matrix at every step with $\mathbf{S}_i = \mathbf{M} + h^2 k \mathbf{J}_i^T \mathbf{J}_i$ (the Hessian of the objective function) and the residual with $\mathbf{r}_i = h\mathbf{f}(\mathbf{x}_i)$. We use a Fletcher-Reeves formula and perform the inner line search in only one iteration - see Algorithm 1.

Note that the NCG method is not necessarily faster than traditional CG linear implicit solvers (we found that it takes roughly 40% more time without optimizations). We can also add back the second derivative term if we want. Also, visually there is no big difference between the two methods. The only advantages you would get with the NCG method are smaller spring elongations and more stability for large time steps. But the main reason for devising the scheme is the similarity with PBD which we further exploit in the next section.

4 CONSTRAINT SPACE

Given that we already know that the regularized projection method is equivalent to implicit integration and that the formulation in the previous section is already very similar to PBD, we would like to transform the system in (12) to one corresponding to fast projection. So by multiplying (12) on the left-hand side by $\mathbf{T} = \frac{1}{hk}\mathbf{A}^{-1}\mathbf{J}$, where $\mathbf{A} = \mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T$, we get:

$$(h^2\mathbf{A} + \varepsilon\mathbf{I})\delta\lambda + \mathbf{c}(\mathbf{x}) = \mathbf{0}, \tag{13}$$

which is precisely the system we need to solve at every iteration of fast projection for the regularized constraints in (9). In order to get this result we made the substitution $\delta\mathbf{v} = h\mathbf{M}^{-1}\mathbf{J}^T\lambda$ which derives from the optimality conditions of the constraint projection optimization problem. To back our claims you can see in Figure 1 that NCG and PBD behave almost the same and very closely to the exact and CG semi-implicit solvers.

We call $\mathbf{T}$ a *constraint space transformation* from configuration space and show that the inverse transform is also possible by multiplying (13) to the left hand side by $\mathbf{Q} = hk\mathbf{M}^{-1}\mathbf{J}^T$. Note that $\mathbf{TQ} = \mathbf{I}_m$ (m - number of constraints) and $\mathbf{QT} = \mathbf{I}_n$ (n - degrees of freedom). We thus found a quick way of switching from one interpretation to the other. Also, by setting $\varepsilon \to 0$ (infinite stiffness) we recover the classic iterative projection formula:

$$h^2\mathbf{A}\delta\lambda + \mathbf{c}(\mathbf{x}) = \mathbf{0}, \tag{14}$$

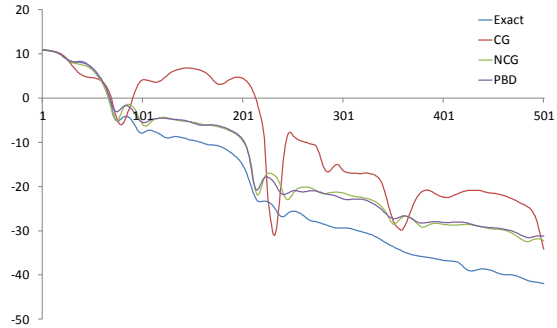


Figure 1: The energy evolution over 500 frames of a 15x15 piece of cloth using NCG (green), PBD (purple) and CG (red) and exact (blue) semi-implicit solvers.

Note though that not all implicit formulations can be converted this way to a constraint based formulation and that is because the methods are equivalent as optimization problems but the numerical methods used may differ in significant ways, e.g. the use of the second derivative. Still their results converge towards the same solution.

5 ENERGY CONSERVATION

Implicit methods in general suffer from artificial numerical dissipation, whether they are used for an elasticity based formulation or a constraint based one. This is usually regarded as a good stability property and the extra damping is considered beneficial by the computer graphics community. Still in many cases like the example of cloth, this integration technique acts like a low-pass filter that removes high frequency motion and thus prevents the formation of high-detail wrinkles and responsive folds.

In the projection methods literature there exist energy preserving solutions like *symmetric projection* [HLW06] or the Velocity Verlet method proposed in [Gol10]. Another popular integration method that can conserve energy exactly is the Implicit Midpoint method [OAW06]. There exist other explicit variational integrators (e.g. Symplectic Euler) with very good energy conservation properties but they suffer from the same time step limitations as any other explicit method.

Given the fact that we are not able to solve the nonlinear equations generated by implicit methods exactly, the

accumulated errors will make Implicit Midpoint much less stable than Implicit Euler. One could alleviate this problem by using techniques suited for explicit methods, e.g. smaller/adaptive time steps or adding in a damping term. Our solution is to employ an integration scheme taken from [Lac07] which gives us more flexibility:

$$\tfrac{1}{h}\mathbf{M}\Delta\mathbf{v} = \mathbf{f}_{ext} + (1-\alpha)\mathbf{f}(\mathbf{x}^{(n)}) + \alpha\mathbf{f}(\mathbf{x}^{(n+1)}), \quad (15)$$

$$\tfrac{1}{h}\Delta\mathbf{x} = (1-\beta)\mathbf{v}^{(n)} + \beta\mathbf{v}^{(n+1)}, \quad (16)$$

where α and β are between 0 and 1 and we can distinguish the following special cases: Explicit Euler ($\alpha=\beta=0$), Implicit Euler ($\alpha=\beta=1$), Implict Midpoint ($\alpha=\beta=\frac{1}{2}$), and Symplectic Euler ($\alpha=0,\beta=1$ or $\alpha=1,\beta=0$).

Moving slightly away from the Implicit Midpoint method and making it more implicit permits us to have low artificial numeric dissipation while still being able to solve the system approximately and obtain a stable, yet responsive simulation. Using the constraint space transformation , i.e. multiplying (15) and (16) on the left hand side by $\mathbf{T}$, we obtain the following regularized projection:

$$(h^2\alpha\beta\mathbf{A}_i + \varepsilon\mathbf{I})\delta\lambda_{i+1} + \alpha\mathbf{c}(\mathbf{x}_i) = \mathbf{0}, \quad (17)$$

where $\mathbf{x}_{i+1} = \mathbf{x}_i + \beta h\mathbf{M}^{-1}\delta\mathbf{f}_{i+1}$, $\mathbf{v}_{i+1} = \mathbf{v}_i + h\mathbf{M}^{-1}\delta\mathbf{f}_{i+1}$ and $\delta\mathbf{f}_{i+1} = \mathbf{J}_i^T\delta\lambda_{i+1}$. Also the integration of the candidate state changes to:

$$\tilde{\mathbf{x}} = \mathbf{x}^{(n)} + h\mathbf{v}^{(n)} + h^2\mathbf{g} + h^2\beta(1-\alpha)\mathbf{M}^{-1}\mathbf{J}^T\lambda^{(n)},$$
$$\tilde{\mathbf{v}} = \mathbf{v}^{(n)} + h\mathbf{g} + h(1-\alpha)\mathbf{M}^{-1}\mathbf{J}^T\lambda^{(n)},$$

where we replaced external force by gravity for brevity and the Jacobian $\mathbf{J}$ is computed at position $\mathbf{x}^{(n)}$.

At the limit $\varepsilon \to 0$ this whole procedure is of course equivalent to projecting the candidate positions using the same system (14) as in fast projection and PBD. The difference appears only in the regularization term which gets replaced by $\varepsilon\alpha^{-1}\beta^{-1}$. The above formulation is also good when stepping both positions and velocities at the same time. Alternatively we could estimate the new velocity only at the end using equation (16):

$$\beta\mathbf{v}^{(n+1)} = \tfrac{1}{h}(\mathbf{x}^{(n+1)} - \mathbf{x}^{(n)}) - (1-\beta)\mathbf{v}^{(n)}$$

We could have reached a similar result using the fast projection formalism in [Gol10] but our constraint space transformation method ensures that we also obtain the correct regularization term. For example, we can apply the same technique to obtain the BDF-2 based projection method presented in [EB08] and find that the new regularization term is $\frac{9}{4}\varepsilon$. Still we prefer our method as we do not need to store previous positions and velocities and, being a one step scheme, it is better suited for non-smoothness. Actually more related to ours is the Newmark scheme [SSB13] for which we can use the same projection method regularized with ε/β, using the Newmark β factor between 0 and $1/2$.

6 DAMPING

Now that we have reduced the amount of artificial damping, we can add back some real damping. Our method will be based on the damping force expression used in [BW98] which is also a special case of the widely used method of *Rayleigh damping* [SSB13]. In order to extend (10) to contain the damping force we need to consider the total force as $\mathbf{f}(\mathbf{x},\mathbf{v}) = \mathbf{f}_e(\mathbf{x}) + \mathbf{f}_d(\mathbf{v})$, i.e. sum of elastic and damping forces:

$$(\mathbf{M} - h^2\mathbf{K} - h\mathbf{D})\delta\mathbf{v} = h\mathbf{f}(\tilde{\mathbf{x}}),$$

where $\mathbf{D} = \frac{\partial\mathbf{f}}{\partial\mathbf{v}} = \nabla_{\mathbf{v}}\mathbf{f}$. This is equivalent to having a damping force:

$$\mathbf{f}_d = d\nabla_{\mathbf{x}}\mathbf{c}(\mathbf{x})\dot{\mathbf{c}}(\mathbf{x}) = d\mathbf{J}^T\mathbf{J}\mathbf{v},$$

where d is the damping coefficient and $\dot{\mathbf{c}}(\mathbf{x}) = \mathbf{J}\mathbf{v}$.

Rayleigh damping makes the approximation $\mathbf{D} = \zeta\mathbf{M} + \gamma\mathbf{K}$, but we will only be using the second term as it makes the derivations simpler and it is only damping along the constraints we are interested in (we can achieve drag friction in other ways). The implicit integration formula (10) now becomes:

$$(\mathbf{M} - h(h+\gamma)\mathbf{K})\delta\mathbf{v} = h\mathbf{f}(\tilde{\mathbf{x}}). \quad (18)$$

If using the approximation in (11) we notice that $\mathbf{f}_d = \gamma\mathbf{K}\mathbf{v}$, where $\gamma = d/k$ is the ratio between the damping and the stiffness coefficients.

We can incorporate this damping force into the optimization formulation using *Rayleigh dissipation functions* [Lac07]. So we transform (18) to constraint space and get the following projection:

$$h(h+\gamma)\mathbf{A}\delta\lambda_{i+1} + \mathbf{e}_i = \mathbf{0}, \quad (19)$$

where $\mathbf{e}_i = \mathbf{c}(\mathbf{x}_i) + \gamma\mathbf{J}\mathbf{v}_i$; the second term is nothing more than the relative velocity along the constraint times γ. So this is a simple way to add more damping along the constraint directions which looks more natural than plain viscous drag in all directions. The downside is that you may have to pay the price of more iterations in order to keep the same amount of constraint violation as before damping. The final formula after adding regularization and energy preservation is:

$$(h\alpha(h+\gamma)\mathbf{A}_i + \varepsilon\mathbf{I})\delta\lambda_{i+1} + \alpha\mathbf{e}_i = \mathbf{0}. \quad (20)$$

As you can see now the projection formula includes both a stiffness parameter ($\varepsilon = 1/k$) and a damping parameter ($\gamma = d/k$) and so the method is fully equivalent to the implicit integration of a damped elastic system. Note that even in the case of infinite stiffness, the damping parameter can remain finite and we get (19).

7 ITERATIVE SOLVERS

As we already mentioned, the most popular methods for solving PBD are nonlinear relaxation methods: Jacobi, Gauss-Seidel or Successive Over Relaxation (SOR). Jacobi for instance has the following update formula: $\delta\lambda_{i+1} = \delta\lambda_i + \mu\mathbf{r}_i$, where $\mu_j = \omega/A_{jj}$, $\omega < 1$ and A_{jj} comes from the diagonal of $\mathbf{A}$. What these methods actually do is they solve each equation and its corresponding unknown separately (local solve) and iterate over all equations for a number of times in order to refine the solution. The local solve formula for one constraint c in the PBD method is:

$$h^2\nabla c(\mathbf{x})^T\mathbf{M}\nabla c(\mathbf{x})\delta\lambda + c(\mathbf{x}) = 0, \qquad (21)$$

where $\mathbf{M}$ and $\mathbf{x}$ correspond to the $s \geq 1$ particles involved in the constraint. We could also solve for more than one constraints simultaneously and we would obtain a system $\mathbf{L}\delta\lambda + \mathbf{c}(\mathbf{x}) = \mathbf{0}$ that we could solve using direct matrix inversion or a direct solver.

Looking more closely at the equation (13) we notice that it is not that different from (14). We can regard the change to the diagonal of the system matrix as a scaling through a relaxation factor as in [Jak01]. As noted in [MHHR07] this factor is highly nonlinear and we are now able to express this exact nonlinear relationship to the linear spring stiffness value: $\omega_j = (1 + (h^2 k_j A_{jj})^{-1})^{-1} < 1$, where k_j is the stiffness of spring j. On the other hand, if we use $\omega > 1$ we obtain SOR which may converge faster and produce better inextensibility, i.e. stiffer cloth for less iterations.

Another application of the equivalence between implicit integration and regularized PBD is to see what happens to the Conjugate Gradient method when we transform from configuration to constraint space. Let us start with the update formula for Steepest Descent:

$$\delta\mathbf{v}_{i+1} = \delta\mathbf{v}_i + \alpha_i\rho_i,$$

where ρ_i is the residual of the system in (12) which is related to the residual $\mathbf{r}_i$ by $\rho_i = \mathbf{T}\mathbf{r}_i$ and $\mathbf{r}_i = \mathbf{Q}\rho_i$, and

$$\alpha_i = \frac{\rho_i^T\rho_i}{\rho_i^T\mathbf{S}\rho_i}.$$

We would like to see how the constraint space transformation affects the update formula in constraint space. So we write α_i in terms of $\mathbf{r}_i$:

$$\alpha_i = \frac{\mathbf{r}_i^T\mathbf{Q}^T\mathbf{Q}\mathbf{r}_i}{\mathbf{r}_i^T\mathbf{Q}^T\mathbf{S}\mathbf{Q}\mathbf{r}_i}.$$

For the nominator we find that $\mathbf{Q}^T\mathbf{Q} = h^2k^2\tilde{\mathbf{A}}$, where $\tilde{\mathbf{A}} = \mathbf{J}\mathbf{M}^{-2}\mathbf{J}^T$ and the denominator is $\mathbf{Q}^T\mathbf{S}\mathbf{Q} = h^2k^3\mathbf{A}\mathbf{S}$. By using the properties of matrices $\mathbf{T}$ and $\mathbf{Q}$ we get that $\mu_i = k\alpha_i$ and by considering infinite stiffness, i.e. $\mathbf{S} \to h^2\mathbf{A}$ when $\varepsilon \to 0$, we get:

$$\mu_i = \frac{\mathbf{r}_i^T(h^2\tilde{\mathbf{A}})\mathbf{r}_i}{\mathbf{r}_i^T(h^2\mathbf{A})^2\mathbf{r}_i}.$$

If we ignore the tilde (or all the masses are 1) we see that we have obtained the formula for the Minimum Residual method which was used in [FM14]. It may be that the method breaks when $\mathbf{M}^{-1}$ is far different from its square and the correct formula needs to be used. Still we think this derivation is a strong argument for why CG does not work in constraint based methods and we need to transform it to a method that looks more like a minimum residual version of CG, e.g. Conjugate Residuals (CR) [Saa03].

8 CLOTH MODEL

The most straightforward application of the presented methods to cloth is through the use of a model made of particles and springs or rigid links connecting them. Such links correspond to a constraint function like $\mathbf{c}(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\| - l_{ij}$, where l_{ij} is the initial rest length of the link and i and j are the indices of the two particles. Using three types of links for stretching, shearing and bending, one can obtain a full model of cloth. Details on how to build such links for quad meshes are given in many papers [Pro96, OAW06]. The main advantage of our method of *soft constraints* is that the stiffness of each constraint can now be expressed naturally as an elastic parameter (related to Young's modulus) instead of using non-linear attenuation factors like in [Jak01] and [MHHR07].

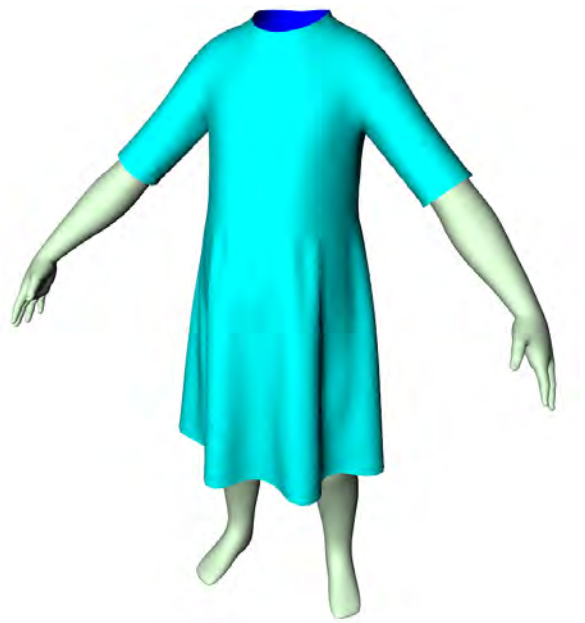


Figure 2: Simulation of a cloth model consisting of 6910 vertices and 13674 triangles using soft constraints

For irregular triangle meshes, one can also build bending links [Erl05], but one cannot distinguish between stretching and shearing links anymore. For quad meshes shearing is simple to express and is usually less stiff than stretching. Using the same stiffness coefficient for shearing as for stretching (infinite in the

case of PBD) leads to locking artifacts, as the cloth does not have enough degrees of freedom to move naturally. Lowering the stiffness value may help with the locking problem, but this causes stretching, which must be avoided at all cost for cloth.

Many other solutions have been proposed for locking [EB08], but we chose to use the model presented in [BW98] which separates out the stretching components (warp and weft) from shearing for each triangle. In general a constraint involving a number of particles implies calculating gradients corresponding to each particle $\nabla_i c(\mathbf{x}) = \frac{\partial c(\mathbf{x})}{\partial \mathbf{x}_i}$. Then we can solve that constraint independently using (21):

$$\delta\lambda = -\frac{c(\mathbf{x})}{\sum_{i=1}^{s} m_i^{-1} \|\nabla_i c(\mathbf{x})\|^2} = -\frac{c(\mathbf{x})}{\xi}. \tag{22}$$

The action of the bending links is dependent on stretching so we might want to use other measures for the curvature of the cloth. We could use directly the constraint between two triangles (4 vertices) defined in [MHHR07], as it expresses the same dihedral angle as in [BW98]. Still we chose to derive our own formulas using (22) and the assumption made in [BW98] that the lengths of the normals remain constant.

9 FINITE ELEMENT METHOD

The continuum formulation in [BW98] is actually a special treatment of the finite element method. The three constraints correspond to the strain components ε_{uu}, ε_{vv} and ε_{uv} that make up the planar symmetric Green-Lagrange strain tensor:

$$\varepsilon(\mathbf{x}) = \tfrac{1}{2}(\nabla\mathbf{w}\nabla\mathbf{w}^T - \mathbf{I}), \tag{23}$$

where $\mathbf{w} : \mathbb{R}^2 \to \mathbb{R}^3$ is a mapping from an undeformed mesh parametrization (u, v coordinates) to deformed vertex positions. Then the actual components are:

$$\varepsilon_{uu} = \tfrac{1}{2}(\mathbf{w}_u^T \mathbf{w}_u - 1), \tag{24}$$

$$\varepsilon_{vv} = \tfrac{1}{2}(\mathbf{w}_v^T \mathbf{w}_v - 1), \tag{25}$$

$$\varepsilon_{uv} = \varepsilon_{vu} = \mathbf{w}_u^T \mathbf{w}_v, \tag{26}$$

where by the subscript of $\mathbf{w}$ we signify partial derivation with respect to to u and v. By considering strain constant over a triangle (linear FEM) we can derive simple formulas for $\mathbf{w}_u$ and $\mathbf{w}_v$ like in [BW98] or [VMTF09].

The integral of the strain energy over a triangle is:

$$U_{\text{fem}} = \tfrac{a}{2}\hat{\varepsilon}(\mathbf{x})^T \mathbf{E}\hat{\varepsilon}(\mathbf{x}), \tag{27}$$

where a is the area of the triangle, $\hat{\varepsilon}^T = (\varepsilon_{uu}, \varepsilon_{vv}, \varepsilon_{uv})$ and $\mathbf{E}$ is a matrix that depends on the Young modulus E and the Poisson ratio ν (or equivalently on the Lamé coefficients) like the one given in [VMTF09] or [TWS07]. Note that the former expresses isotropic elasticity while the latter expresses orthotropic elasticity, i.e. different stiffness along warp and weft directions.

We use (27) to derive the true constraint function using the regularization framework as in [SLM06]:

$$\mathbf{c}(\mathbf{x}) = a^{\frac{1}{2}}\hat{\varepsilon}(\mathbf{x}). \tag{28}$$

The resulting three constraints (c_u, c_v, c_s) are are similar to the ones in [BW98] and their gradients form the Jacobian $\mathbf{J}_{\text{fem}} = (\nabla c_u^T, \nabla c_v^T, \nabla c_s^T)$. Note that in the most rigorous approach the area of the triangle $a(\mathbf{x})$ is also varying and its derivative should also be considered. We chose not to do so in our computations, but alternatively we could add an extra area constraint. Some authors use area and volume constraints together with edge constraints to improve on mass-spring soft body models [THMG04].

Now we can formulate the regularization condition:

$$a^{\frac{1}{2}}\hat{\varepsilon}(\mathbf{x}) + \mathbf{E}^{-1}\lambda = \mathbf{0}. \tag{29}$$

In the end we can apply the block local solve formula from Section 7, which is equivalent to other StVK linear FEM approaches like the one in [VMTF09]. We choose to apply this block approach only for the stretch components together, as the shear stress component is related only through a diagonal term to strain, and thus decoupled from the normal directions. The resulting 2x2 local linear system for the two stretching constraints is:

$$(h^2\mathbf{A} + \tilde{\mathbf{E}}^{-1})\delta\lambda + \hat{\varepsilon}(\mathbf{x}) = 0,$$

where in the case of isotropic materials

$$\tilde{\mathbf{E}}^{-1} = \frac{1}{E\sqrt{a}}\begin{pmatrix} 1 & \nu \\ -\nu & 1 \end{pmatrix}.$$

Notice that we divided equation (29) by the constant area term $a^{1/2}$ and obtained a CFM matrix that contains all the relevant continuous material parameters: Young's modulus, Poisson ratio and the triangle area (discretization measure). We can also add damping through the Rayleigh damping technique presented in Section 6 or the projection in Section 5 for better energy conservation. In the end we obtain a very accurate (iterations permitting) and physically correct model for simulating thin nonlinear elastic materials like the one in [VMTF09] based only on constraints (Figure 3).

10 RESULTS

We implemented a cloth simulator solely on a modular constraints architecture using C++ (single threaded). Depending on the level of accuracy or performance the user can choose between different constraint types, e.g. links or FEM triangles for stretching and shearing, different types of bending constraints, static collision or

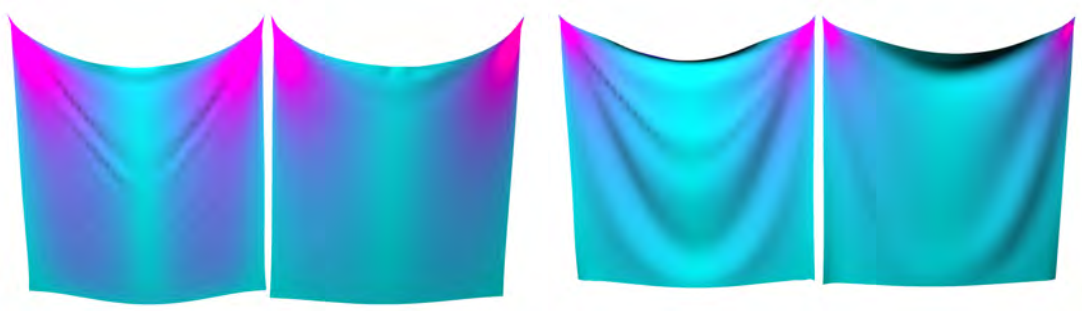


Figure 3: Two snapshots of a side by side real-time simulation of two 40x40 cloth pieces with the same Young's modulus E: regularized FEM constraints (left) and soft links (right); superimposed in purple is the strain map. FEM offers more realistic folds and the strain is better distributed throughout the cloth.

self collision constraints etc. Note that all these constraints are treated in the same solver loop. Regularization was implemented by modifying the diagonal of the system matrix using equation (13) or as described in Section 7. The resulting scheme can be modified to use the projection in Section 5 or we could add more damping (Section 6). These options can be added depending on the needs of the simulator and we denote them collectively as soft constraint methods for enhancing PBD. This also is why we do not provide any pseudo-code and hope that the readers will assemble themselves the simulator of choice.

Given our simulator is fully constraint based our collision response techniques are the same as the ones used in [Jak01, MHHR07] and for self collision we adapted the methods in [BFA02]. Friction is treated more accurately by being solved at every contact iteration in a similar fashion to cone complementarity programming [TA10]. We implemented cloth-mesh collisions by testing for triangle-point and edge-edge intersections between two triangles. For acceleration we used AABB trees for both the static mesh (pre-computed) and the cloth (rebuilt at every frame). A similar approach was used for accelerating self-collisions too.

We tested simulations mostly visually looking for obvious artifacts like jittering or instability. Our most common test scenario was a piece of cloth hanging by two corners, falling from a horizontal or vertical position, with different parameters or tessellation. Given the multitude of methods used and the differences between them it is hard to find a metric that measures well the quality of the simulation. We opted to measure the total energy - kinetic and potential (gravitational and elastic) and no damping, and chart its evolution in time (Figure 4). The NCG solver behaves well and has good convergence, but decays non-monotonically. The regularized PBD method is smoother, dissipates energy slower, but the Gauss-Seidel solver is less accurate. Energy preserving projection with $\alpha = \beta = 0.6$ offers even slower energy decay while the higher energy line is due to the kinetic energy of the oscillation. This is a good energy preserving property but it looks jittery and unnatural and we may need to add extra (non-artificial) damping. The closer we get to $\alpha = \beta = 0.5$ the more horizontal the graph becomes, i.e. full energy conservation, but this is dangerous territory for stability. We get the same results with a Symplectic Euler integrator with very small time steps and a small damping factor.

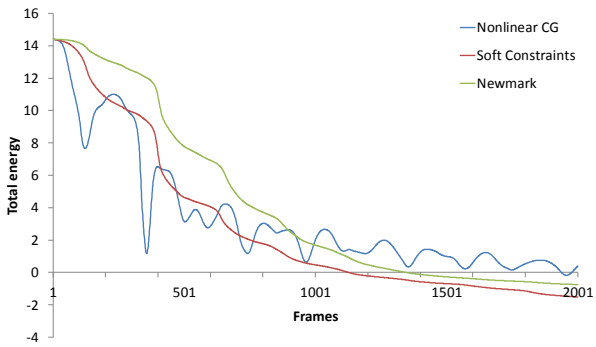


Figure 4: Total energy evolution in time for the simulation of a 10x10 rubber cloth (k = 2 N/m, 25 iterations) using NCG implicit integration (blue), regularized PBD (red) and regularized energy preserving projection (green).

For our damping method we measured the total energy minus the elastic potential in order to give a clearer picture of the velocity reduction (Figure 5). As you can see a damping factor of $\gamma = 10h$ gives a significant energy dissipation compared to soft projection (or PBD just as well). Reaching this level of dissipation so quickly is not possible using the method we compared against, i.e. reducing the relative velocity along the constraint direction (basically velocity projection). We set for the energy preserving projector $\alpha = \beta = 0.55$ and $\gamma = h$ in order to obtain as little artificial damping as possible while at same time damping the simulation just a bit less than PBD would normally do.

All simulations were performed in real time at 60 Hz, i.e. under 16 ms of computation time depending on

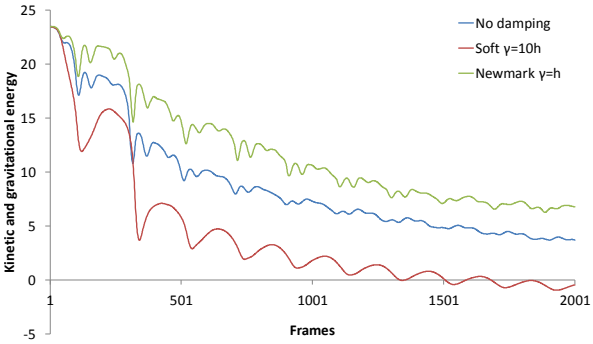


Figure 5: Damping response for the simulation of a 40x40 piece of cloth ($k = 2000$ N/m, 25 iterations) using regularized PBD (blue), aggressive damping (red) and slightly damped energy preserving projection (green).

cloth size, and with lower framerate for the dress in Figure 2 (up to 20 ms or more for the solver only).

11 CONCLUSIONS

We have shown that implicit integration of elastic systems is equivalent as an optimization problem to fast projection. Based on the analogy to PBD we derived a Nonlinear Conjugate Gradient implicit solver. Its drawback is that it is using an approximated force Jacobian but this is compensated by running more than one inexact Newton iterations.

After developing a method of switching between the two representations (configuration and constraint space) we proved that the regularized PBD method (soft constraints) replicates elastic behavior. We also showed how to preserve energy better or how to dissipate more when solving constraints. Note though that the viscous drag term of the Rayleigh damping matrix cannot be treated implicitly in this framework. Also, one can use a parallel version of the Conjugate Residuals algorithm to speed up the simulation. Finally we showed that accurate FEM simulation of cloth using constraints is possible and is no different from implicit integration. We believe that these are new and useful results for PBD.

We hope to use the Kawabata evaluation system in the future for real fabric modeling like in [VMTF09]. We also intend to optimize our simulator using parallel algorithms for multi-core and GPGPU.

12 ACKNOWLEDGMENTS

We would like to thank Framebreed animation studio for the 3D model of the dress and the body in Figure 2. We also want to thank Lucian Petrescu and Dongsoo Han for their input. Part of the research presented in this paper was supported by the Sectoral Operational Program Human Resources Development, 2014-2015, of the Ministry of Labor, Family and Social Protection through a Financial Agreement between University POLITEHNICA of Bucharest and AM POS DRU Romania, project ID 132395.

13 REFERENCES

[AH04] M. Anitescu and G. D. Hart. A Constraint Stabilized Time-Stepping Approach for Rigid Multibody Dynamics with Joints, Contact and Friction. Int. J. Numer. Meth. Eng., 60(14):2335–2371, 2004.

[Bar94] D. Baraff. Fast Contact Force Computation for Nonpenetrating Rigid Bodies. In ACM SIGGRAPH 1994 Papers, pages 23–34, 1994

[BW98] D. Baraff and A. Witkin. Large Steps in Cloth Simulation. In ACM SIGGRAPH 1998 Papers, pages 43–54, 1998.

[BKLS95] E. Barth, K. Kuczera, B. Leimkuhler, and R. D. Skeel. Algorithms for constrained molecular dynamics. J. Comp. Chem, 16:1192–1209, 1995.

[BKCW14] J. Bender, D. Koschier, P. Charrier, and D. Weber. Position-based simulation of continuous materials. Computers & Graphics, 44:1–10, 2014.

[BML*14] S. Bouaziz, S. Martin, T. Liu, L. Kavan, and M. Pauly. Projective dynamics: Fusing constraint projections for fast simulation. ACM Trans. Graph., 33(4):154:1–154:11, 2014.

[BFA02] R. Bridson, R. Fedkiw, and J. Anderson. Robust Treatment of Collisions, Contact and Friction for Cloth Animation. In ACM SIGGRAPH 2002 Papers, pages 594–603, 2002.

[BMF03] R. Bridson, S. Marino, and R. Fedkiw. Simulation of clothing with folds and wrinkles. In Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pages 28–36, 2003.

[Cat10] E. Catto. Soft Constraints: Reinventing the Spring. Game Developer Conference, 2010.

[CK02] K.-J. Choi and H.-S. Ko. Stable but responsive cloth. In Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '02, pages 604–611, 2002.

[EEH00] B. Eberhardt, O. Etzmuß, and M. Hauth. Implicit-explicit schemes for fast animation with particle systems. In In Eurographics Computer Animation and Simulation Workshop, pages 137–151, 2000.

[EB08] E. English and R. Bridson. Animating developable surfaces using nonconforming elements. ACM Trans. Graph., 27(3):66:1–66:5, 2008.

[Erl05] K. Erleben et al. Physics-based Animation. Charles River Media, 2005.

[Erl07] K. Erleben. Velocity-based Shock Propagation for Multibody Dynamics Animation. ACM Trans. Graph., 26, 2007.

[FM14] M. Francu and F. Moldoveanu. An improved jacobi solver for particle simulation. In VRIPHYS 14, pages 125–134, 2014.

[Gol10] A. R. Goldenthal. Implicit Treatment of Constraints for Cloth Simulation. PhD thesis, 2010.

[GHF*07] R. Goldenthal, D. Harmon, R. Fattal, M. Bercovier, and E. Grinspun. Efficient Simulation of Inextensible Cloth. In ACM SIGGRAPH 2007 Papers, 2007.

[HLW06] E. Hairer, C. Lubich, and G. Wanner. Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations. Springer, 2006

[HET01] M. Hauth, O. Etzmuss, and U. Tübingen. A high performance solver for the animation of deformable objects using advanced numerical methods. In In Proc. Eurographics 2001 (2001), pages 319–328, 2001.

[HCJ*05] M. Hong, M.-H. Choi, S. Jung, S. Welch, and J. Trapp. Effective constrained dynamic simulation using implicit constraint enforcement. In Robotics and Automation, ICRA 2005, pages 4520–4525, 2005.

[Jak01] T. Jakobsen. Advanced Character Physics. In Game Developers Conference Proceedings, pages 383–401, 2001.

[Lac07] C. Lacoursière. Ghosts and machines: regularized variational methods for interactive simulations of multibodies with dry frictional contacts. PhD thesis, Umeå University, 2007.

[Lan70] C. Lanczos. The Variational Principles of Mechanics. Dover Publications, 1970.

[LBOK13] T. Liu, A. W. Bargteil, J. F. O'Brien, and L. Kavan. Fast simulation of mass-spring systems. ACM Trans. Graph., 32(6):214:1–214:7, 2013.

[MM13] M. Macklin and M. Müller. Position based fluids. ACM Trans. Graph., 32(4):104:1–104:12, July 2013.

[MSW14] D. L. Michels, G. A. Sobottka, and A. G. Weber. Exponential integrators for stiff elastodynamic problems. ACM Trans. Graph., 33(1):7:1–7:20, Feb. 2014.

[MHHR07] M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff. Position Based Dynamics. J. Vis. Comun. Image Represent., 18(2):109–118, 2007.

[MSJT08] M. Müller, J. Stam, D. James, and N. Thürey. Real time physics: Class notes. In ACM SIGGRAPH 2008 Classes, pages 88:1–88:90, 2008.

[OAW06] S. Oh, J. Ahn, and K. Wohn. Low damped cloth simulation. Vis. Comput., 22(2):70–79, 2006.

[Pro96] X. Provot. Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behavior. In In Graphics Interface, pages 147–154, 1996.

[Saa03] Y. Saad. Iterative Methods for Sparse Linear Systems. Society for Industrial and Applied Mathematics, 2nd edition, 2003.

[SLM06] M. Servin, C. Lacoursière, and N. Melin. Interactive simulation of elastic deformable materials. In SIGRAD 2006 Conference Proceedings, pages 22–32, 2006.

[She94] J. R. Shewchuk. An Introduction to the Conjugate Gradient Method Without the Agonizing Pain. Technical report, Pittsburgh, PA, USA, 1994.

[SSB13] F. Sin, D. Schroeder, and J. Barbic. Vega: Nonlinear fem deformable object simulator. Comput. Graph. Forum, 32(1):36–48, 2013.

[Smi06] R. Smith. Open dynamics engine v0.5 user guide. 2006.

[Sta09] J. Stam. Nucleus: Towards a unified dynamics solver for computer graphics. In Computer Aided Design and Computer Graphics, pages 1–11, 2009.

[SD06] A. Stern and M. Desbrun. Discrete geometric mechanics for variational time integrators. In ACM SIGGRAPH 2006 Courses, SIGGRAPH '06, pages 75–80, 2006.

[ST96] D. Stewart and J. C. Trinkle. An Implicit Time-Stepping Scheme for Rigid Body Dynamics with Coulomb Friction. Int. J. Numer. Meth. Eng., 39:2673–2691, 1996.

[TA10] A. Tasora and M. Anitescu. A Convex Complementarity Approach for Simulating Large Granular Flows. J. Comput. Nonlinear Dynam., 5, 2010.

[THMG04] M. Teschner, B. Heidelberger, M. Muller, and M. Gross. A versatile and robust model for geometrically complex deformable solids. In Proceedings of the Computer Graphics International, pages 312–319, 2004.

[TPS09] B. Thomaszewski, S. Pabst, and W. Straßer. Continuum-based strain limiting. Comput. Graph. Forum, 28(2):569–576, 2009.

[TWS07] B. Thomaszewski, M. Wacker, and W. Straßer. Advanced topics in virtual garment simulation – part 1, 2007.

[VMTF09] P. Volino, N. Magnenat-Thalmann, and F. Faure. A simple approach to nonlinear tensile stiffness for accurate cloth simulation. ACM Trans. Graph., 28(4):105:1–105:16, 2009.

[WN99] S. J. Wright and J. Nocedal. Numerical optimization, volume 2. Springer New York, 1999.

Hybrid ROI-Based Visualization of Medical Models

Lazaro Campoalegre
Trinity College Dublin
College Green
Dublin 2, Ireland
campoall@tcd.ie

Isabel Navazo
Universitat Politècnica de Catalunya
C. Jordi Girona 31
08034 Barcelona, Spain
isabel@cs.upc.edu

Pere Brunet
Universitat Politècnica de Catalunya
C. Jordi Girona 31
08034 Barcelona, Spain
pere@cs.upc.edu

ABSTRACT

There is an increasing interest on tele-medicine and tele-diagnostic solutions based on the remote inspection of volume data coming from multimodal imaging. Client-server architectures meet these functionalities. The use of mobile devices is sometimes required due to the portability and easy maintenance. However, transmission time for the volumetric information and low performance hardware properties, make quite complex the design of efficient visualization systems on these devices. In this paper, we present a hybrid approach which is based on regions of interest (ROIs) and on a transfer-function aware compression scheme. It has a good performance in terms of bandwidth requirements and storage needs in the client device, being flexible enough to represent several materials and volume structures in the ROI. Clients store a low-resolution version of the volume data and ROI-dependent high resolution segmented information. Data must be only sent whenever a new ROI is requested, but interaction in the client is autonomous - without any data transmission - while a certain ROI is inspected. A benchmark is presented to compare the the proposed scheme with three existing approaches, on two different volume data models. The results show that our hybrid approach is compact, efficient and scalable, with compression rates that decrease when the size of the volume model increases.

Keywords

Volume rendering, client-server, mobile devices, medical data, region of interest, ray-casting, volume data compression

1 INTRODUCTION

Recently, several important research areas in three-dimensional techniques for multimodal imaging have appeared. Applications include neurological imaging for brain surgery, tissue characterization, medical school teaching, plastic surgery and others. At the same time, scientists are more familiarized with three-dimensional structures reconstruction from Two-dimensional images.

The reconstruction of a volumetric model is generally achieved by using a voxel representation of datasets. According to the structure to be highlighted during the visualization, a transfer function is applied to assign color and opacity to the density value which represents the structure properties.

The handling of three-dimensional information requires efficient systems to achieve fast data transmission and interactive visualization of high quality images. Client-server applications allow these functionalities. Sometimes the use of mobile devices is necessary due to the portability and easy maintenance. However, transmission time for the volumetric information and low performance hardware properties, complicate the design of efficient visualization systems on these devices.

The main contribution of our work is a Hybrid visualization approach that inherits the advantages of some previous algorithms like the ones presented in [1] and [2], while keeping a good performance in terms of bandwidth requirements and storage needs in client devices. The scheme is flexible enough to represent several materials and volume structures in the Region of Interest (ROI) at high resolution and very limited information transmission cost.

2 PREVIOUS WORK

Client-server architectures have grown in popularity. Mobile devices as well as desktop computers can both function as clients requesting and receiving information over the network. Many authors have published research results in the remote volume visualization area. However there is still scarce specific bibliography for volume visualization in mobile devices. The majority of the proposals use known algorithms like Ray-Casting, 2D Textures, and isosurface modeling to render volume data. In order to compensate limitations in low performance devices or to reduce costs, the number of client-server schemes have been proposed.

In some client-server approaches the dataset is compressed on the server side and sent to the client where

the transfer function is applied after decompression and before the rendering of the recovered data. Moser and Weiskopf [3] proposed a 2D texture-based method which uses compressed texture atlas to reduce interpolation costs. Nogera et al. [4] proposed a webGL application to visualize very large 3D volumes by using multi-texturing to encode volumetric models on a set of RGBA 2D textures. A recent application developed by Balsa et al. [5] allows to interact with volume models using mobile devices hardware. Their scheme is not compressing the volume data.

In other schemes, the transmitted data is a compressed image, the transfer function is applied at the beginning of the pipeline, followed by a 2D rendering on a texture, all done on the server side. A compressed image is sent to the client where decompression and image rendering takes place. This scheme is frequently named "Thin Clients" [6]. The idea in multiresolution model schemes [7] is to render only a region of interest at high resolution and to use progressively low resolution when moving away from that region. Both bricking and multiresolution approaches [8] need a high memory capacity on the CPU for storing the original volume dataset. Moreover, bricking requires a high amount of texture transfers as each brick is sent once per frame; multiresolution techniques have been built for CPU purposes and its translation to GPUs is not straightforward due to the required number of texture accesses.

Preprocessing of data is also a useful technique, as it ensures the reduction of the information, combined with different techniques for quantization, encoding and multiresolution representation [8].

Efficient schemes require optimized algorithms to reduce and send data through the network. The algorithms must achieve the maximum compression possible while allowing an easy decompression in the client side, where sometimes hardware and memory constraints decrease performance [8].

Wavelet transforms offer considerable compression ratios in homogeneous regions of an image while conserving the detail in non-uniform ones. The idea of using 3D wavelets for volume compression was introduced by Muraki [9]. Ihm and Park [10] proposed an effective 3D 16^3-block-based compression/decompression wavelet scheme for improving the access to random data values without decompressing the whole dataset. Guthe et al. [11] proposed a novel algorithm that handles a hierarchical wavelet representation where decompression takes place in GPU.

Some techniques advocate the use of hybrid region-based volume rendering, by applying different shading algorithms inside the volume model [12], or by implementing multiresolution region-based schemes [1]. Luo *et al.* [13] developed a technique for focusing on a user-driven ROI while preserving context information. The approach uses a distance function to define the region of interest. This function controls voxel opacity, exploits silhouette enhancement and non-photorealistic shading.

In this paper, we propose a hybrid framework that exploits the use of standard transfer functions as an alternative to compress volume dataset. Our scheme is a transfer function-aware scheme for client/server techniques. It combines Wavelet-preprocessed volume data to reduce information outside the ROI, and highlighted segmented data in regions of interest (ROI), (Gradient Octree shceme). From the best of our knowledge this possibility has not been considered by any of the described approaches in this previous work.

3 OVERVIEW OF THE APPROACH

Let us assume that we are interested in inspecting a volume data model V which is too large in terms of network transmission and/or client storage facilities. Wavelet compression algorithms like the ones presented in [1, 10, 11] are able to support block-based regions of interest (ROIs). Other approaches like Gradient Octrees [2] can be rendered with advanced illumination models and at a higher visual quality level. Gradient Octrees are specific data structures for multiresolution volume datasets. Gradient Octrees $G(V)$ include an specific data structure S and a compact data array D. The octree structure S can be sent to the client devices in a lossless way with only one bit per node, whereas data is compacted to 3 Bytes per octree node, including material information and volume gradients. Both approaches, however, have advantages and drawbacks:

(I) TF-aware wavelet compression schemes succeed in sending to the clients a very limited amount of information in the areas outside the region of interest (ROI). High quality volume information in the ROI is also compact [1], because the 3D texture is smaller and restricted to the blocks in the ROI area. Ray-casting visualization in the client can use compact 3D textures which are suitable for many client devices. The main drawback of this approach, however, is twofold. First, changing the transfer function requires sending a new version of the compress volume model to the client, and second, it is not well suited for illumination computations that would require too many texture accesses.

(II) Approaches like Gradient Octrees overcome these limitations by supporting multiple transfer functions and materials and by precomputing gradients on the server. They support advanced illumination models, thus achieving a higher visual quality level. However, they are not direct candidates for ROI-based visualization paradigms, as their low level volume representations present a flat-face appearance with poor gradients. These representations at coarse tree levels are

well suited for progressive transmission but they perform worse than similar-quality low-level wavelet reconstructions.

Our hybrid scheme inherits the best of both approaches. In this case, apart from the volume model V, the user must supply a set of transfer functions $\{TF_k\}$ and selects one of them as a canonical transfer function. The server starts by computing a Gradient Octree $G(V)$ from V and for the set $\{TF_k\}$, also computing the quantified representation $W(V)$ of the wavelet transform of V with the canonical transfer function, as described in Section 4. $G(V)$ encodes materials and gradients only in the subset of voxels of V which are relevant to same of the transfer functions in $\{TF_k\}$.

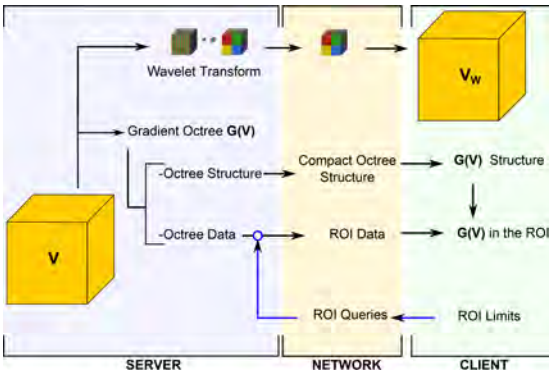


Figure 1: Overview of the proposed scheme, showing the preprocess on the server, the data transfer through the network and the data structures in the client device

Users at the client side can interactively define regions of interest, ROIs. Information over the network can be classified into *static information* (being send only once per volume model) and *dynamic information*. Dynamic information must be re-sent whenever the ROI is redefined by the user. Static information is compact, including $W(V)$ and a set of arrays defining the tree structure of $G(V)$. In cases where the size of the volume model V is too large and the volume data at the deepest level of $G(V)$ does not fit into the client's CPU memory, the portion of this data belonging to the ROI is generated from the octree data on demand, as *dynamic information*, whenever the user asks for a different ROI.

In the client side, a low-resolution volume model V_W is reconstructed by de-quantizing and computing a few inverse wavelet steps in each block. Let us note as V_R the subvolume corresponding to the ROI. A two-level ray-casting rendering algorithm in the client GPU (Section 5) succeeds at showing a high-quality Gradient Octree rendering in V_R together with a visualization of V_W in the parts of the volume outside the ROI, also supporting a number of interaction facilities.

The corresponding compression and decompression algorithms are detailed in Section 4.

4 COMPRESSION AND DECOMPRESSION ALGORITHMS

We start by computing the wavelet transform $W(V)$ of the volume model V and its gradient octree $G(V)$ on the server, Figure 1. We use a localized, block-based transform with a previous smoothing step to achieve local behaviour and a better compression rate. We assume standard piecewise linear transfer functions [**?**]. By considering these transfer functions, we virtually segment the volume V in as many regions as linear segments defined by the $\{TF_k\}$ functions. Voxels with a density d such that the opacity of $\{TF_k\}$ is zero for all k, belong to null regions and are simply represented by a null code. Our implementation uses a block size of 16 together with a 4-steps Haar transform, being rather efficient in compression while supporting block-aware interaction paradigms in the client. As already mentioned, the wavelet information that is sent over the network to the client is a low resolution volume model V_W, obtained by computing a few w_l inverse wavelet steps in each block. Observe that in the usual case of $w_l = 2$, the size of the information in V_W will always be lower than 1/64 of the size of the initial model V. The low-resolution model for $w_l = 2$ is compressed more than a 98.5%. In what follows, we will use the term *compression rate* to name the relative size of the compressed model, which in this case is 1.5%.

The gradient octree $G(V)$ information includes the octree structure and the octree data. Creating $G(V)$ involves three compression steps. The first is transfer-function aware and uses V and the set of $\{TF_k\}$ to compute an Edge Volume model $VE(TF)$ which only encodes voxels that are relevant to the transfer functions $\{TF_k\}$. Non-relevant voxels in $VE(TF)$ are assigned a Nil value. On a second step, we compress gradient information to a total of three bytes per Grey tree node (including material information) in a set of data arrays, one per octree level, [2]. We use a GPU-oriented encoding of the proposed hierarchical data structure with explicit volume gradient information in octree nodes, to avoid gradient computations during GPU ray-casting. The final Gradient Octrees representation, shown in Figure 2, consists on a small volume model V_{32} with pointers and two sets of per-level arrays, O_l and D_l. For the sake of clarity, octree levels in Figure 2 and in what follows will be identified by their resolution. The example shown in Figure 2 corresponds to the complete octree representation of a volume V of resolution $r = 512$, with gradient and materials stored in the data array D_{512}. Data arrays of coarser octree levels (D_{256}, D_{128}, D_{64} and D_{32}) store gradient and materials data of Grey octree nodes at these levels. In short, the octree structure S of $G(V)$ consists of the pointers volume V_{32} and the set of per-level arrays O_l. The octree data D of $G(V)$ includes the set of per-level arrays D_l.

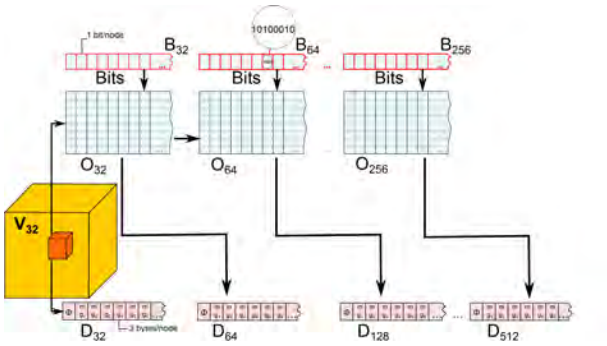


Figure 2: Encoding Gradient Octrees

For transmission purposes, it is possible to compact the Gradient Octree structure (not the volume data) in a lossless way. Instead of sending the arrays O_l, we send a set of arrays B_l with one byte per parent node as shown in red in Figure 2. Note that this is equivalent to store and transmit only one bit per node: the components in B_l simply represent the type of each child in one bit (0 if Nil, 1 if Grey). Compressed arrays B_l are computed on the server and sent to the client. For every received level, the client is able to generate a B_l-driven, increasing sequence of indexes to create a local copy of the array of indexes to child nodes O_l (for a detailed discussion, see [2]). Note that O_l indexes point simultaneously to D_{2l} and to O_{2l}. The volume V_{32} is sent to the GPU as a 3D texture, whereas arrays O_l and D_l are encoded as 2D and 1D textures. In short, we succeed in sending the tree structure S in a lossless way and with only one bit per node, through a sequence of compact arrays B_l. Moreover, compressing gradients and materials in three bytes is efficient, supports GPU decompression and suffers from a very limited loss in visual quality.

Although ROI-dependent localizations of the octree structure S could be defined [14], we have observed that the corresponding compression improvements (mainly in the information over the network) are negligible. In our present implementation we have therefore considered a hybrid model consisting of the low-resolution volume V_W, the gradient octree structure S and ROI-dependent octree data D. This hybrid model information is sent from the server to the client (or clients) in two parts: (I) The static information is sent only once, at the beginning of the interaction session. It consists on the low-resolution volume V_W, the $32 \times 32 \times 32$ pointers volume V_{32} of the computed Gradient Octree, the set of arrays B_l which encode the S octree structure and the materials look-up table of the Gradient Octree, Figure 1. The size of this last table is very small and we will not consider it in our compression computations. (II) The dynamic information is sent on demand whenever the client changes the ROI. The client sends a query with the new ROI limits (bounding box) and the server generates and sends a subset D_R of the data arrays D of the Gradient Octree, as we know in advance that only voxels in the ROI will be retrieved and rendered, Figure 1.

In our present implementation we assume that users are only interested in gradient octree data at the deepest octree level r, as lower resolutions are already shown outside the ROI. This makes the whole process easier, as we can just send a compact D_r array containing only those voxels with a non-Nil gradient value in the deepest octree level. This results in a very compact data transmission. We compute and keep a temporal, ROI-dependent version of the pointers volume V_{32} which we name VR_{32}. VR_{32} has Nil pointers outside the ROI and sequential pointers for the Grey nodes inside the ROI. The textures VR_{32} and O_k are now ROI-dependent, and must be recomputed in the client from the $G(V)$ Structure (see Figure1) whenever the ROI is changed during the interaction, with a very efficient algorithm which only involves array traversal and counting.

5 RENDERING AND INTERACTION IN THE CLIENT DEVICES

Reconstruction of any of the blocks within the non-ROI partion of the volume can be performed at one, two, three or four wavelet levels. The four-level reconstruction of a block generates a full piece of $16 \times 16 \times 16$ voxels that represent the corresponding part of the volume. Reconstructions of the same block at three, two or one levels generate pieces of $8 \times 8 \times 8$, $4 \times 4 \times 4$ or $2 \times 2 \times 2$ voxels, representing the same part of the volume at lower resolutions.

A usual interactive session starts by inspecting the whole volume model at a low resolution. In this case, all blocks are usually reconstructed at one or two levels, the corresponding 3D Texture is sent to the client GPU and ray-casting rendered. Observe that the size of this 3D texture, in the case of two reconstruction levels, is 1/64 of the size of the original volume model V.

Alternatively, the user can decide to inspect the whole volume model at a low resolution (two levels of reconstruction, for instance) with the ROI showing predefined structures at maximum level of detail by ray-casting the Gradient Octree. To achieve this last interactive visualization, two structures, one for the non-ROI volume (3D texture) and the other for the ROI volume (Gradient Octree), are sent to the client GPU where an adaptive ray-casting algorithm is performed, as detailed below. Since the whole model is available at the client side, rotation and zooming operations can be autonomously performed in the client without any further transmission from the server. If a region of the model needs to be detailed, the Gradient Octree can be displayed on demand.

We use a standard ray casting algorithm in the client GPU, the main difference with the classical algorithms

being that rays traverse a low resolution 3D texture representing the whole volume. In the point samples along the ray that do not belong to the ROI, the ray-casting uses density values from the low-resolution model V_W. Samples in the ROI retrieve densities from a virtual volume with the same resolution R as V_R, instead of traversing V_R itself. In Figure 3, an example with resolution $R = 512$ is shown. Ray-casting proceeds as usual by advancing along rays r from the observer with a uniform sampling of the volume along r. Then, for each sample s of r addressing a virtual voxel (i, j, k) in the ROI, its volume information is found in D_R.

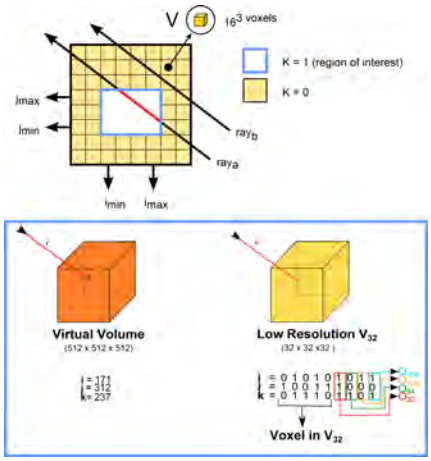


Figure 3: The block structure of the model, a region of interest (in white) and the octree-based ray-casting.

Ray casting within the ROI is based on the octree addressing properties. The octree search of any virtual voxel (i, j, k) is directly driven by the base-2 representation of i, j and k, as shown in Figure 3. In this case, their first 5 binary digits point to the corresponding voxel in the low-res texture V_{32}. The index i_{32} found in this V_{32} voxel element points to the low resolution data in D_{32} (which we don't use if a higher resolution is required) and also to the array of its eight child indexes in O_{32}. A well-known property of binary octree subdivision ensures that next "three bit columns" in the binary representation of i, j, k are in fact child indexes s_l. Son indexes point to deeper octree levels and are able to drive the octree traversal to the right element in D_R containing data in the virtual volume voxel. Subtree traversal from the low-res voxel in V_{32} to the virtual voxel data is based on the recursion equation,

$$i_{2l} = O_l[i_l][s_l] \tag{1}$$

for l=32, 64, 128, .. $R/2$

The final index i_R points to the high-res data in D_R, but tree traversal can stop earlier if the virtual voxel is void and any index i_l in the chain is found to be zero.

Observe that only virtual voxels in V_R in the ROI will be addressed. This means that the client must only store a restriction of $G(V)$ in V_R. In our present implementation we initially send the whole octree structure of $G(V)$ to the client, but high-resolution data D_R (restricted to V_R) is only sent on demand when the user changes the ROI

Let's assume that ray r is crossing voxel $i = 171$, $j = 312$, $k = 237$ in the virtual volume of the ROI, Figure 2. In this case, the octree search starts in the voxel $(10, 19, 14)$ of V_{32} and is then driven by four child indexes: $s_{32} = 7$, $s_{64} = 1$, $s_{128} = 4$ and $s_{256} = 5$ which recursively generate the indexes i_{64}, i_{128}, i_{256} and i_{512}. Reaching the deepest level information in a Gradient Octree of resolution $R = 512$ involves a maximum of six texture queries, to V_{32}, O_{32}, O_{64}, O_{128}, O_{256} and finally to D_{512}.

After retrieving high-res data in D_{512}, materials and the gradient vector are decompressed on the fly in the GPU. Obviously, everything also works when lower resolution virtual volumes are considered.

6 RESULTS AND DISCUSSION

To perform a complete comparative study , we selected the following accessible frameworks:

VrMed Viewer, an integration of libraries and functionalities, designed to achieve interactive visualization in PCs and Virtual Reality Systems, using a GPU-based Ray-casting algorithm [15].

Volume Viewer, an Android based application [5], implemented to run on mobile devices. Allows interactive visualization of models with a transfer function editor with easy handling. In this case, the whole volume model is sent to the client device.

VrMed-Thin Client The approach is based on [16]. It achieves remote visualization of volume models with basic user interaction tools in mobile devices. A server running VRMed Viewer on Linux operative system, renders images which are sent to the client through the wireless. Clients generate control commands as OpenGL parameters which are sent to the server using a TPC/IP socket.

Tables 1 and 2 show a comparison of [2], [1], our technique (**Hybrid Approach**), and the previously described schemes, using two models: The skull model with a $256 \times 256 \times 112$ resolution and the thorax model with a 512^3 resolution. Density values are in the rang [0...255], hence each voxel is codified using only 1 byte. Table rows show for each scheme whether multi-resolution and progressive transmission is allowed and the compression rate achieved for each case. Both tables also show the size of the transmitted data trough the network and the client requirements to perform volume rendering followed by an estimation of the average frame rates in two cases: rendering in the PC server and rendering in the client (mobile device).

Figure 4 shows some snapshots of the interaction with the hybrid skull model. In all cases, two wavelet reconstruction steps ($wl = 2$) have been used without lighting computation in the low resolution area: In (a), (b) and (d) the Wavelet have been computed by using a skin tranfer function, whereas the image in (c) represents a bone transfer function both in the low resolution area and in the Region of Interest.

Some snapshots of the interaction with the hybrid thorax model are shown in Figure 5. In all cases two wavelet reconstruction steps ($wl = 2$) have also been applied without lighting computation, in the low resolution area: ROI showing ribs and lungs (a), internal gases and lungs (b) and skin, ribs and lungs (d). The snapshot in (e) shows the alveoli in the ROI, magnified in (f) by interacting with zoom and a section plane. In these cases wavelets are precomputed after applying to the model a TF covering all structures in the low resolution area. The snapshot in (c), shows a TF for bones visualization in both, the low resolution area, and the ROI. Image in (d) shows a zoom-in of (c) for showing up the quality of the hybrid model. The server application runs on PC with 6 GB of RAM, Intel Core 2 Duo at 3.16 GHz and a client with 4 GB of RAM, Intel Core 2 Duo at 3.06 GHz and Nvidia GeForce GTX z80. Client tests were performed on the HTC One smartphone whith a screen resolution of 1080 x 1920 pixels 2 GB RAM and an Adreno 320 Graphics processor.

A ROI-based visualization has been considered in the Wavelets-based scheme and in the Hybrid approach, while in the rest of columns, the whole volume V has been rendered at a uniform resolution. This is valid in both cases (tables 1 and 2). Zoom has been adjusted in a way that the total amount of rendered ROI pixels in the application viewport is a 25% of the total of viewport pixels. The amount of ROI pixels in the viewport is relevant, as it measures the total amount of required high-quality casted rays during ray-casting rendering.

Compression rates correspond to the amount of data sent over the network, and relate this amount to the total memory requirements of the volume models, which are 7.4 MBytes in Table 1 and 128 MBytes in the case shown in Table 2. In contrast to the previous schemes, our techniques allow multi-resolution rendering with progressive transmission of volume data. For the Wavelet based approach, the presented figures on the amount of data over the network represent the necessary information to reconstruct four levels of wavelet compression, $w_l = 4$. The compression rate in this case is between 21% and 32%.

In case of the Gradient Octree approach, data includes the octree structure plus material and gradient information at its deepest, maximum resolution level. In the Hybrid scheme, the *information over the network* represent both the necessary data to reconstruct two levels of wavelet compression for the low resolution model and the nodes representing the Gradient Octree leaves in the selected region of interest (ROI). This approach also requires a client GPU being able to manage 3D textures. The compression rate in this case is between 20% and 22%, with an average frame rate in the mobile device between 7 and 16 fps.

The proposal in this paper **Hybrid approach** results in a compression rate which is between 4% and 18%, with an average frame rate in the mobile device between 8 and 16 fps when $w_l = 2$. It also requires a client GPU being able to manage 3D textures.

The VrMed viewer is presented for comparison purposes. Some of the parameters in the tables do not apply to this case, as VrMed is a stand-alone application without network transmission. The average frame rates, 48 and 20 fps, are obviously higher than those in the previous cases but these figures show that our proposed approaches are performing within reasonable efficiency limits.

The Thin Client based approach sends a maximum of 0.18 MB of data through the network per frame during an interactive session with a single client (of course, the total amount of transmitted data depends on the number of interaction frames). This is due to the fact that the technique requires the transmission of rendered images from the server when the user interacts with the model in the client side. This fact makes this scheme network dependent, with framerates which decrease in network congestion cases. We have observed that our thin-client implementation becomes useless when the number of clients is above 8. On the other side, this scheme does not require sophisticated client GPUs, as clients must only decompress and show pre-rendered images. This can be an advantage for basic client devices, but result in an under-utilization of client GPUs in the case of most present devices. The asterisks in the *Thin client* column in tables 1 and 2 mean that data sizes are per frame sizes. The compression rates obviously depend on the number of transmitted frames.

The Hybrid approach is specially well suited in the case of large models. The comparison between tables 1 and 2 show that this is a scalable scheme, with compression rates that decrease when the size of the volume model increases. The corresponding frame rates are larger than in the case of Gradient Octrees, being of the same order of magnitude than Thin Clients.

Comparing Thin clients to Wavelets, Gradient Octrees and the Hybrid approach, we can define the break-even interaction period as the number of frames required to have an equivalent amount of information sent over the network. Break-evens are computed as the ratio between the size of the compressed model as sent over the network in our approaches and the size of a single Thin Client frame image. In the case of the skull model

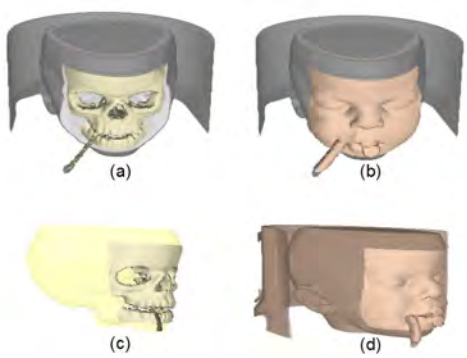


Figure 4: Hybrid Visualization. Interaction with the hybrid skull model. ROI size: $(128 \times 64 \times 64)$.

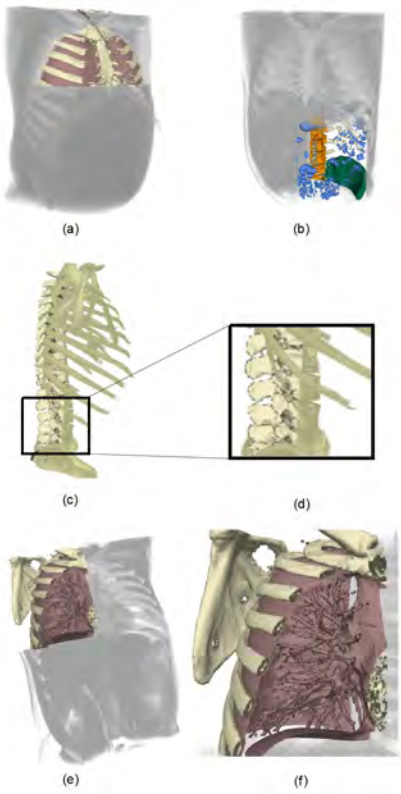


Figure 5: Hybrid Visualization. Interaction with the hybrid thorax model. ROI size: $416 \times 224 \times 224$ (a), $256 \times 160 \times 256$ (b), $96 \times 128 \times 480$ (c), and $256 \times 512 \times 256$ (d), (e) and (f).

in Table 1, this break-even is 11 frames for Wavelets, 21 frames for Gradient Octrees and 11 frames for the Hybrid approach.

In the case of the thorax model in Table 2, the break-even is 51 frames for Wavelets, 79 frames for Gradient Octrees and 17 frames for the Hybrid approach. By considering the number of frames per second in each case, we can conclude that the information we are sending is equivalent to the total information sent by the Thin Client approach during an interaction period in between 1 and 10 seconds. In the case of the Hybrid approach, break-evens are 11 frames and 17 frames, meaning this Hybrid scheme outperform Thin Clients in interaction cases longer than around 20 frames.

Thin Clients can also be compared with the presented approach in terms of frame rates. Frame rates depend on the network bandwidth, the present approach being better than Thin Client approaches in geographic regions with a limited bandwidth. In fact, the presented proposal can be specially useful in world regions with limited network infrastructures but requiring fast access to 3D medical data, like non-urban areas.

The Volume Viewer approach as presented in the last column of both tables does not require sophisticated client GPUs, as clients are rendering stacks of 2D textures. Frame rates in the client are reasonable. The main drawback in this case, however, is the amount of information being sent over the network, which makes it unusable in the case of large volume models.

The proposed hybrid scheme allows interactive inspection by rotating and zooming volume models. Users are able to select ROI portions of the visualized model, as well as choosing a transfer function from the set of transfer functions ($\{TF_k\}$) inside the the selected ROI. Interface options include also, planar section selection and offsets structures visualization in front of the selected section plane. Users can also choose the resolution of the low resolution region, by reconstructing models, by using one, two or three wavelet reconstruction steps.

	Wavelets Sheme	Gradient Octrees	Hybrid Approach	VrMed Viewer	VrMed Thin Client	Volume Viewer
Multi-resolution	✓	✓	✓	no	no	no
Progressive transmission	✓	✓	✓	no	no	no
Compression rate (%)	32	22	18	-	$*$	100
Data over the network(MB)	1.91	3.8	2.04	-	0.18*	7.3
Client requirements	3D Tex	3D Tex	3D Tex	-	Image	Stack 2D Tex
Frame rate (PC version)	52	24.12	46.53	48.24	48.24	-
Frame rate (mobile)	24.2	-	16.32	-	20.23	17.0

Table 1: Comparison between the approach presented in this paper (blue column) and some previous schemes for remote visualization of volume models. Study of the Skull model with a resolution of $256 \times 256 \times 112$ and 7.3 MB of size.

	Wavelets Sheme	Gradient Octrees	Hybrid Approach	VrMed Viewer	VrMed Thin Client	Volume Viewer
Multi-resolution	✓	✓	✓	no	no	no
Progressive transmission	✓	✓	✓	no	no	no
Compression rate (%)	21	20	4.2	-	$*$	100
Data over the network(MB)	16.4	27	5.3	-	0.32*	128
Client requirements	3D Text	3D Tex	3D Tex	-	Image 2D Tex	Stack
Frame rate (PC version)	14	10.31	18.03	20.34	20.34	-
Frame rate (mobile)	13.20	-	8.07	-	20.23	-

Table 2: Comparison between the approach presented in this paper (blue column) and some previous schemes for remote visualization of volume models. Study of the Thorax model with a resolution of 512^3 and 128 MB of size.

7 CONCLUSIONS & FUTURE WORK

We have proposed a Hybrid approach that inherits the advantages of the algorithms presented in [1] and [2] while keeping a good performance in terms of bandwidth requirements and storage needs in client devices. Information over the network consists on static information (being only set once) and dynamic information. Dynamic information must be re-sent whenever the ROI is redefined by the user. The complexity (memory and data transmission requirements) of the static and dynamic information has been discussed. The main conclusion is that the hybrid scheme is flexible enough to represent several materials and volume structures in the ROI area at a very limited static and dynamic information transmission cost.

The Hybrid approach has been proved to be specially well suited in the case of large models. The presented experimental tables show that the Hybrid approach is a scalable scheme, with compression rates that decrease when the size of the volume model increases. Corresponding frame rates are larger than in the case of Gradient Octrees, being of the same order of magnitude than Thin Clients. Our compression results are better than similar client server schemes for volume rendering, and compare favourably to Marching Cubes based approaches. While these last schemes must send an average of three triangles per voxel in segmented volumes, the presented approach only sends three bytes/voxel. We consider that our approach may enrich the user experience during the inspection of volume medical models in these low performance devices.

ACKNOWLEDGMENT

This work has been partially supported by the Project TIN2013-47137-C2-1-P of the Spanish Ministerio de Economía y Competitividad.

8 REFERENCES

[1] Lázaro Campoalegre, Pere Brunet, and Isabel Navazo. Interactive visualization of medical volume models in mobile devices. *Personal Ubiquitous Comput.*, 17(7):1503–1514, 2013.

[2] Lázaro Campoalegre, Pere Brunet, and Isabel Navazo. Gradient octrees: A new scheme for remote interactive exploration of volume models. *Proc. of the CAD/Graphics*, 2013.

[3] Manuel Moser and Daniel Weiskopf. Interactive volume rendering on mobile devices. In *Vision, Modeling, and Visualization VMV*, volume 8, pages 217–226, 2008.

[4] Jose Maria Noguera and Juan Roberto Jiménez. Visualization of very large 3d volumes on mobile devices and webgl. In *WSCG international conference on computer graphics, visualization and computer vision*, 2012.

[5] Marcos Balsa and Pere Vázquez. Practical volume rendering in mobile devices. In *Advances in Visual Computing*, pages 708–718. Springer, 2012.

[6] Klaus Engel and Thomas Ertl. Texture-based volume visualization for multiple users on the world wide web. In *Virtual Environments*, pages 115–124. Springer, 1999.

[7] Eric LaMar, Bernd Hamann, and Kenneth I. Joy. Multiresolution techniques for interactive texture-based volume visualization. In *IEEE Visualization*, pages 355–361, 1999.

[8] Enrico Gobbetti, José Antonio Iglesias Guitián, and Fabio Marton. Covra: A compression-domain output-sensitive volume rendering architecture based on a sparse representation of voxel blocks. In *Computer Graphics Forum*, volume 31, pages 1315–1324, 2012.

[9] Shigeru Muraki. Volume data and wavelet transforms. *Computer Graphics and Applications, IEEE*, 13(4):50–56, 1993.

[10] Insung Ihm and Sanghun Park. Wavelet-based 3D compression scheme for interactive visualization of very large volume data. *Computer Graphics Forum*, pages 3–15, 1999.

[11] Stefan Guthe, Michael Wand, Julius Gonser, and Wolfgang Straßer. Interactive rendering of large volume data sets. In *Visualization, 2002. VIS 2002. IEEE*, pages 53–60. IEEE, 2002.

[12] Jianlong Zhou, Manfred Hinz, and Klaus D Tönnies. Hybrid focal region-based volume rendering of medical data. In *Bildverarbeitung für die Medizin 2002*, pages 113–116. Springer, 2002.

[13] Yanlin Luo, José Antonio Iglesias Guitián, Enrico Gobbetti, and Fabio Marton. Context preserving focal probes for exploration of volumetric medical datasets. In *Proceedings of the International Conference on Modelling the Physiological Human*, pages 187–198, 2009.

[14] Lazaro Campoalegre. Contributions to the interactive visualization of medical volume models in mobile devices. In *PhD Thesis, UPC*, 2014.

[15] Eva Monclús, José Díaz, Isabel Navazo, and Pere-Pau Vázquez. The virtual magic lantern: An interaction metaphor for enhanced medical data inspection. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pages 119–122, 2009.

[16] C. Rezk-Salama, K. Engel, M. Bauer, G. Greiner, and T. Ertl. Interactive volume on standard pc graphics hardware using multi-textures and multi-stage rasterization. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware*, pages 109–118, 2000.

Joint Bilateral Mesh Denoising using Color Information and Local Anti-Shrinking

Oliver Wasenmüller
oliver.wasenmueller@dfki.de

Gabriele Bleser
gabriele.bleser@dfki.de

Didier Stricker
didier.stricker@dfki.de

DFKI GmbH - German Research Center for Artificial Intelligence
Augmented Vison Department
Trippstadter Str. 122
67663 Kaiserslautern, Germany

ABSTRACT

Recent 3D reconstruction algorithms are able to generate colored meshes with high resolution details of given objects. However, due to several reasons the reconstructions still contain some noise. In this paper we propose the new Joint Bilateral Mesh Denoising (JBMD), which is an anisotropic filter for highly precise and smooth mesh denoising. Compared to state of the art algorithms it uses color information as an additional constraint for denoising; following the observation that geometry and color changes often coincide. We face the well-known mesh shrinking problem by a new local anti-shrinking, leading to precise edge preservation. In addition we use an increasing smoothing sensitivity for higher numbers of iterations. We show in our evaluation with three different categories of testdata that our contributions lead to high precision results, which outperform competing algorithms. Furthermore, our JBMD algorithm converges on a minimal error level for higher numbers of iterations.

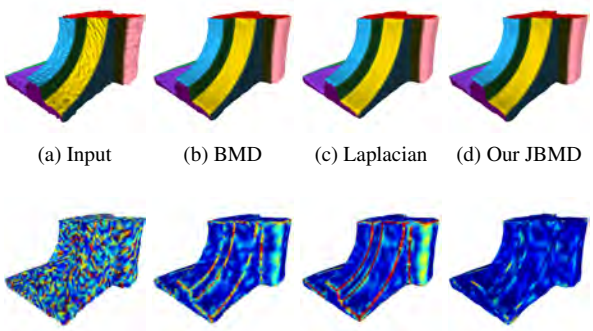
Keywords

Mesh Denoising, Smoothing, Fairing, Joint Bilateral Filter, Local Anti-Shrinking, Color Information.

1 INTRODUCTION

Many applications, such as urban planning, industrial measurement or human anthropometry, require reconstructed 3D models of the respective objects with very high precision. The traditional approach is to acquire these models by laser scanners, since they promise high quality results. However, they are expensive, impractical to use and contain still some noise. Meanwhile 3D reconstruction algorithms, such as e.g. [Agi, FP10, NIH+11, SSC14], are able to generate colored mesh models from devices like standard color cameras and\or depth cameras, which are widely spread, cheap and easy to use. These reconstructions contain color information together with high-resolution details, but also suffer from noise in their 3D geometry. To get rid of this noise, several (mostly iterative) methods for mesh denoising were proposed in the literature. Sometimes they are also referred to as smoothing, filtering or fairing methods. They use directly the 3D geometry or derived measures, like distances or normals of the mesh, to estimate new vertex positions. However, none of them explicitly uses the color information provided by e.g. one of the above mentioned algorithms. Thus, we present in this paper a new anisotropic method called Joint Bilateral Mesh Denoising (JBMD), which uses - besides geometric information - the color information as an additional constraint for edge preserving denoising.

Another well-known problem of mesh denoising are shrinking effects. They occur mostly in curves regions of the mesh and are caused by homogeneous shifts of vertices in a neighborhood into one major direction. Current approaches try to compensate that ef-



(a) Input (b) BMD (c) Laplacian (d) Our JBMD

Figure 1: Comparison of different mesh denoising algorithms for the *fandisk* mesh. Top row: meshes. Bottom row: color-coded error distribution.

fect, whereas they focus more on visually appealing results than on precision. Thus, in our JBMD algorithm we propose a new approach to avoid this effect by a precise local anti-shrinking. Furthermore, many current algorithms suffer from their high dependence on the number of iterations. Therefore our new algorithm increases the denoising sensitivity per iteration leading to constantly low errors. A short overview of our algorithm is also given in the following video: `https://youtu.be/odm8kr2rKPA`

Summarizing, the main contributions of our proposed JBMD algorithm are:

- explicit usage of color information as an addition constraint for denoising,
- precise local anti-shrinking and
- increasing denoising sensitivity.

The remainder of this paper is organized as follows: Section 2 gives an overview of existing methods for mesh denoising. The proposed JBMD algorithm is motivated and explained in detail in Section 3, while it is evaluated regarding precision in Section 4. The work is concluded in Section 5.

2 RELATED WORK

Image and mesh denoising is an ongoing research topic in the image processing and computer vision community.

Often, mesh denoising methods are related to image denoising approaches. State of the art approaches in image denoising include methods such as anisotropic diffusion [PM90], total variation [ROF92], wavelet denoising [Don95], robust diffusion [BSMH98], bilateral filter [TM98] and joint bilateral filter [KCLU07, HSJS08]. In particular the joint bilateral filter uses, similar to our new approach, the color information as an additional constraint.

State of the art algorithms for mesh denoising are amongst others Laplacian [Fie88, Tau95, VMM99] and bilateral [FDCO03, JDD03, ZFAT11] mesh denoising (BMD) methods. Laplacian mesh denoising is an iterative isotropic procedure, where the new vertex positions are directly calculated from the positions of the neighboring vertices. In contrast, bilateral mesh denoising is an iterative edge preserving anisotropic approach. New vertex positions are estimated from the vertex's neighborhood, where the influence of neighboring vertices depends on their distance and on their offset to the tangent plane. Parts of this approach are also used for our new algorithm.

A general and well-known problem of mesh denoising is that the mesh shrinks in convex regions with each application of the particular algorithm, which is a huge problem especially for iterative approaches. [Tau95] solves this problem by alternating shrinking and expansion steps. Admittedly the precision of this approach depends heavily on the geometry of the particular mesh [DMSB99]. Another common approach, which is e.g. used in the Bilateral Mesh Denoising [FDCO03], is to preserve the volume of the mesh by a global correction step as proposed in [DMSB99]. The algorithm estimates the volume V^n of a mesh after the n-th iteration by the sum of volumes of all ordered pyramids centered at the origin and with a triangle of the mesh as base. Each vertex of the mesh is then scaled by the factor β, which is defined by

$$\beta = \left(\frac{V^0}{V^n}\right)^{\frac{1}{3}}, \tag{1}$$

to achieve the original volume V^0. However, as mesh shrinkage occurs only in convex regions contrary to flat regions, a global correction has indeed appealing effects but is not precise. Thus, we propose in this paper a precise local shrinkage correction.

3 METHOD

In this paper we propose the Joint Bilateral Mesh Denoising (JBMD), which is a filtering method for meshes using local neighborhoods. The method can be subdivided into two parts: the denoising itself and the subsequent local anti-shrinking.

The main idea of our new denoising algorithm is related to image processing, namely motivated trough the Joint Bilateral Filter (JBF) [KCLU07]. This anisotropic edge-preserving filter is often used to denoise depth images by using color images as additional constraints. The main idea is to compute a new depth value as a weighted average of surrounding depth values, where the weights depend on their deviation in position (space) and color value (range). The assumption of JBF are coherent depth and color discontinuity, meaning that edges in the color image coincide with edges in the depth image and vice versa. This coherence assumption was validated in many image processing publications [KCLU07, WBS15] and we show in Section 4 that it also holds for meshes.

The intention of our new local shrinkage correction is - contrary to alternating [Tau95] or global [DMSB99] correction - to adjust vertex positions only where shrinkage effects occurred. This effect arises only in convex regions, whereas flat regions are not affected. We observed that the weighted mean signed shift of vertices in the neighborhood, which were estimated by the denoising in the first step, equalize in noisy flat regions, whereas in convex (and thus shrunk) regions it is a precise local measure for a shrinkage correction.

Like in many other mesh denoising algorithms [FDCO03, ZFAT11] we estimate in our JBMD new

vertex positions v'' in a mesh by shifting along the normal direction n. This has the positive effect that irregularities in the resulting mesh are avoided. Our two-step algorithm can be described by

$$v'' = v + (x' \cdot n) - (x'' \cdot n), \tag{2}$$

where $(x' \cdot n)$ is the denoising part, $(x'' \cdot n)$ the correction part and $x'|x''$ refer to the magnitude of the shift. The algorithm is illustrated in detail in Figure 2 and defined in the following:

Algorithm 1 Joint Bilateral Mesh Denoising (JBMD) for the m-th iteration

Require: Vertex v, Normal n
1: $\{q_i\}$ = neighborhood(v)
2: $sum, norm, sum', norm' = 0$
3: **for all** i **do**
4: $d_i = \|v - q_i\|$
5: $o_i = \langle n, v - q_i \rangle$
6: $c_i = (v.r - q_i.r)^2 + (v.g - q_i.g)^2 + (v.b - q_i.b)^2$
7: $w_i^d = exp(-d_i^2/2\sigma_d^2)$
8: $w_i^o = exp(-o_i^2/2(\sigma_o \cdot \lambda^m)^2)$
9: $w_i^c = exp(-c_i/2\sigma_c^2)$
10: $sum += (w_i^d \cdot w_i^o \cdot w_i^c) \cdot o_i$
11: $norm += w_i^d \cdot w_i^o \cdot w_i^c$
12: **end for**
13: $x' = sum/norm$
14: $v' = v + x' \cdot n$
15: **for all** i **do**
16: $d_i' = \|v' - q_i'\|$
17: $w_i^d = exp(-(d_i')^2/2\sigma_d^2)$
18: $sum' += w_i^d \cdot x_i'$
19: $norm' += w_i^d$
20: **end for**
21: $x'' = sum'/norm'$
22: $v'' = v' - x'' \cdot n$
23: **return** v"

In the first step of our algorithm (line 3-14) the new position v' of a vertex v is estimated as a weighted average of neighboring vertex position q_i, where the weights depend on three influencing factors: distance (w_i^d), offset (w_i^o) and color difference (w_i^c). For computing the distance d_i between a vertex v and neighboring vertex q_i, the geodesic distance on the smooth surface would be the correct measure. However, for efficiency reasons we approximate d_i using the Euclidean distance in line 4, since [FDCO03] demonstrated already a sufficient impact. The offset o_i is defined as the distance of vertex q_i to the tangent plane of vertex v. The intention of using this offset o_i is that neighboring points in flat regions should have a higher influence than in convex or edge regions. As described in line 5, o_i can be easily estimated using the dot product. The last influence factor is the color difference c_i between a vertex q_i and vertex v, which is estimated in line 6. To map the influence factors d_i, o_i and c_i to weights w_i^d, w_i^o and w_i^c, we use the Gaussians of lines 7-9. The final shift x' of vertex v is the normalized weighted sum of offsets o_i of neighboring vertices q_i.

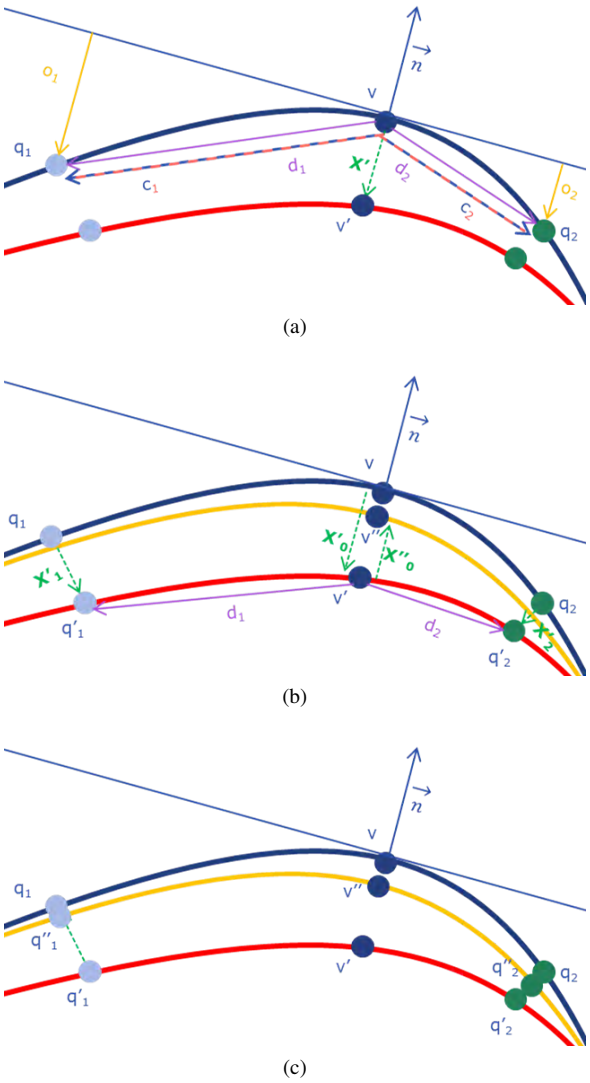


Figure 2: Joint Bilateral Mesh Denoising (JBMD) applied to vertex v: (a) Denoising step ($v \rightarrow v'$). (b) Shrinking correction step ($v' \rightarrow v''$). (c) Final result.

In the second step of our algorithm (line 15-22) we correct the position of a vertex v' due to possible shrinking effects. As already mentioned before, we observed that the weighted mean signed shift x'' of vertices in the neighborhood, which were estimated by the denoising in the first step, is a good local measure for a shrinkage correction. For the estimation of the weights for x'' we use the distance d_i' between a vertex q_i' and v' together with a mapping function (line 16-17) similar to the first step of the algorithm. The weighted mean signed shift x'' is calculated by summing up the weighted signed sifts x_i' (line 18) and normalizing afterwards (line 21).

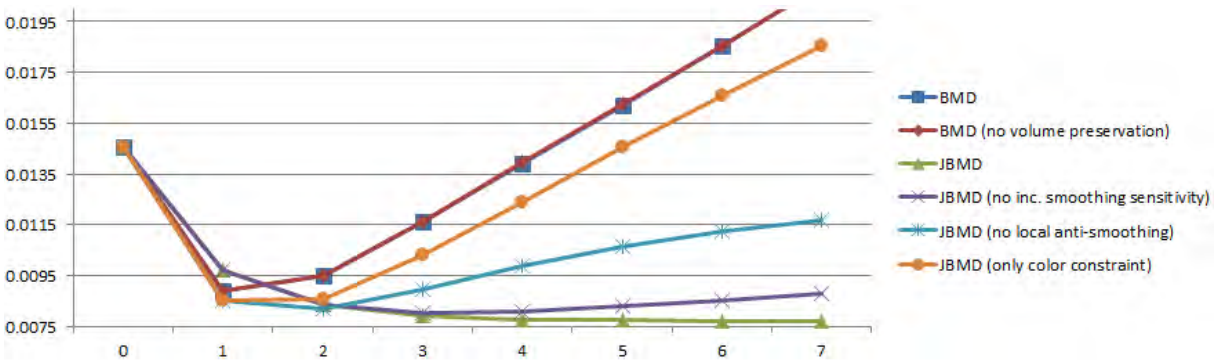


Figure 3: Mean errors of the *fandisk* mesh for different numbers of iterations. Quantitative comparison of different features of our new JBMD algorithm and comparison against BMD.
Dark blue: BMD [FDCO03]. Red: BMD without volume preservation. Green: Our new JBMD. Purple: JBMD without increasing smoothing sensitivity. Light blue: JBMD without local anti-shrinking. Orange: JBMD without increasing smoothing sensitivity and without local anti-shrinking.

The described JBMD algorithm is applied locally for each vertex of the mesh. However, vertices at a boundary of a mesh do not have a well defined neighborhood (line 1). In our algorithm we define the size of the neighborhood as a fixed number k. The neighborhood of a vertex v is then defined by the k closest vertices q_i. Obviously, the shape of our neighborhood changes from vertex to vertex, but since the distance d_i between vertices v and q_i is an influencing parameter, this artifact has negligible influence.

Our JBMD algorithm is - like many other [FDCO03, ZFAT11] - an iterative approach. In the first iteration major noise is eliminated, whereas with higher number of iterations the overall level of noise decreases. Thus, we consider this aspect by an increasing smoothing sensitivity. In our JBMD algorithm the noise influences the result via the offset o_i, whereas the corresponding mapping function depends on σ_o. Therefore, we decrease the parameter σ_o by a constant factor λ with each iteration; leading to constantly low error.

4 EVALUATION

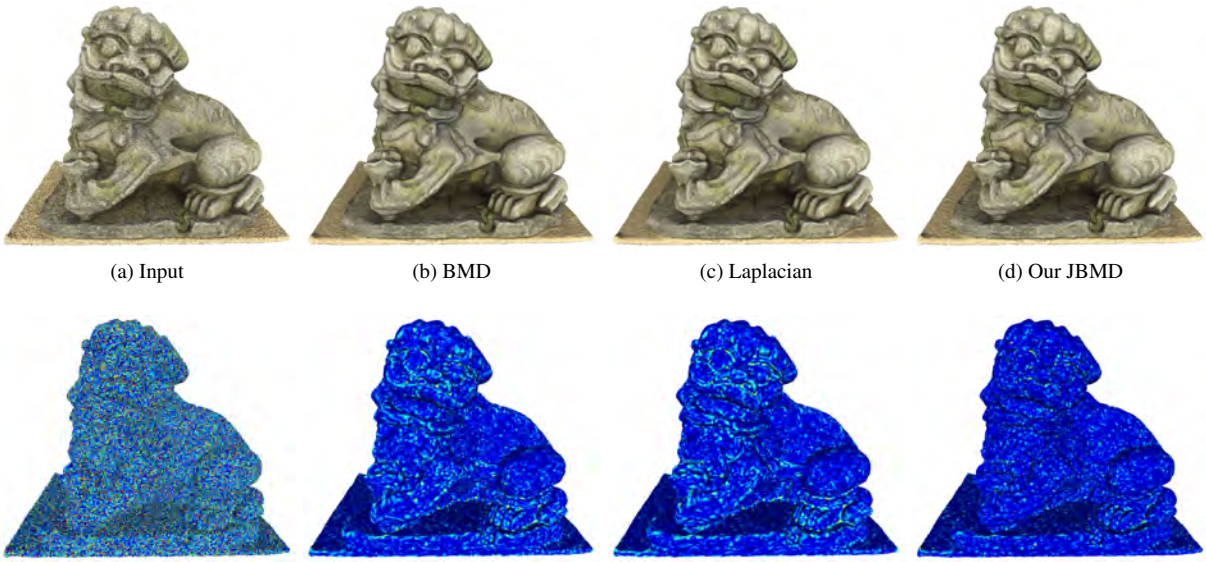
In this section we benchmark our JBMD algorithm by comparing it to competing algorithms, namely Laplacian Denoising and Bilateral Mesh Denoising (BMD). These algorithms are described in more detail in Section 2. All methods - including ours - depend on some parameters. For our JBMD algorithm these are σ_d, σ_o σ_c and λ. Thus, we run each algorithm with a huge number of possible parameter combinations to detect the optimal setting. All results (Figures, diagrams, etc.) shown in this paper are generated with optimal parameter settings and numbers of iterations. For our JBMD algorithm we used the parameter settings of Table 1 for the given datasets. Note, these parameters depend highly on the mean vertex distance (MVD) of the given mesh. According to our experiments $MVD \approx 2\sigma_d \approx 4\sigma_o$ can be used as a rough guideline for setting the parameters.

Unfortunately, a groundtruth comparison on real world data is very difficult, since no datasets are available, which provide both real noisy data and real denoised data. Thus, in the recent literature it is common to use precise models of an object and generate the noise on it synthetically. For this paper we decided to use three categories of testdata in our evaluation.

The first category are colored synthetic meshes with sharp edges together with an artificially noisy version of this mesh. We use here the well-known *fandisk* mesh, where each part of the surface has another color. Furthermore, we add a Gaussian noise, where the standard deviation is roughly half of the vertex distance. The second category of testdata are highly precise reconstructions of real objects acquired by a camera-projector-system [KNRS13]. We also added here a Gaussian noise with a standard deviation of approximately half vertex distance. The meshes used in this paper are the *lion* and *allegorie* reconstructions. The third category of testdata are reconstructed meshes, which are generated by standard cameras and Agisoft PhotoScan [Agi]. We use in this paper the heads of two persons: *person 1* and *person 2*. These reconstructions include partially strong noise due to the lack of characteristic features. Obviously, for these reconstructions no ground truth is available, but they are a real world scenario, where the application of a mesh denoising algorithm is required.

	MVD	σ_d	σ_o	σ_c	λ
fandisk	0.1897	0.1	0.05	30	0.8
lion	0.4401	1.0	0.5	30	0.6
allegorie	0.0003	0.0008	0.0004	35	0.7
person 1	0.0030	0.01	0.005	20	0.7
person 2	0.0041	0.01	0.005	20	0.7

Table 1: Parameter settings in the evaluation of our new JBMD algorithm for the given datasets with specified mean vertex distance (MVD).



(a) Input (b) BMD (c) Laplacian (d) Our JBMD

Figure 4: Comparison of different mesh denoising algorithms for the *lion* mesh. Top row: meshes. Bottom row: color-coded error distribution.

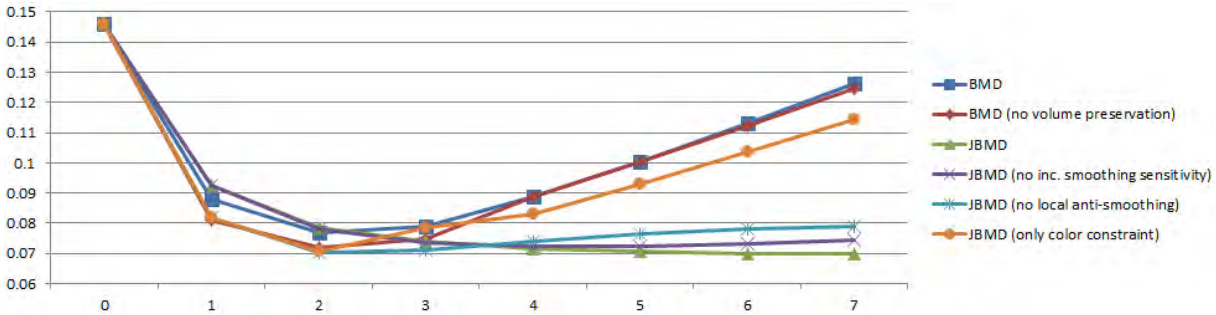


Figure 5: Mean errors of the *lion* mesh for different numbers of iterations. Quantitative comparison of different features of our new JBMD algorithm and comparison against BMD.

As a quality measure for denoising we use the mean error (ME), which is defined for a denoised mesh X and a ground truth mesh G by

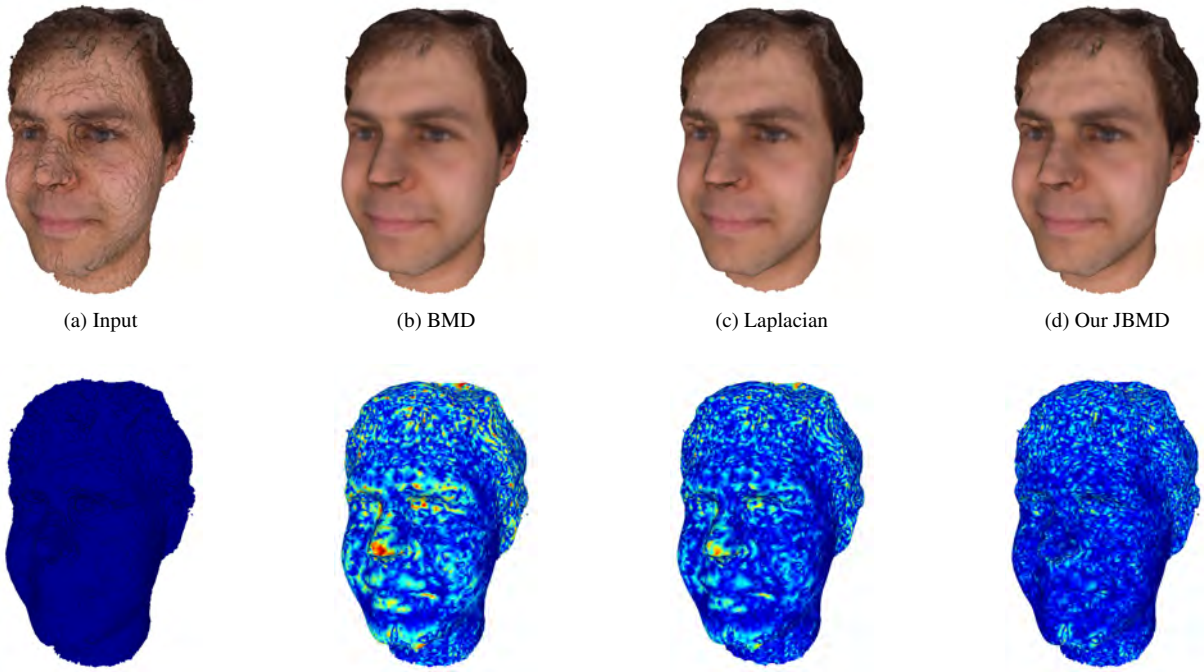
$$ME(X,G) = \frac{1}{n}\sum_{i=1}^{n} \|x_i - g_p\| \qquad (3)$$

$$x_i \in X; g_p \in G; \forall i \; p = \arg\min_p \|x_i - g_p\|,$$

where n is the number of vertices in the mesh X. More intuitive, it is defined by the mean distance of each vertex in the mesh X to the respective closest vertex in the ground truth mesh G.

First of all we evaluate our algorithm with the *fandisk* mesh in Figure 1, which has ideal preconditions for our JBMD, since all sharp edges coincide with color changes. From a visual point of view, all three mesh denoising algorithms provide smooth results without visible noise. However, they differ strongly in their precision, as visible in the color-coded error distribution in the bottom row. The blue color indicates low errors, whereas red represents high errors. Both BMD and Laplacian denoising have imprecise vertices at the edges of the mesh, whereas our JBMD has only some minor inaccuracy. Figure 3 depicts the mean error of the *fandisk* mesh depending on the number of iterations. Our JBMD has the lowest error and converges in particular on this low error level. If our JBMD is used without the increasing smoothing sensitivity, the mean error increases again from the fourth iteration on. If we switch off our shrinking correction, we achieve better results for a small number of iterations. This is caused by the inhibiting effect of the shrinking correction, since it reverts the denoising to some extent. However, for larger numbers of iterations superior results can be achieved with our new local anti-shrinking. Looking at the effects of using the color information as an additional parameter, we see that our JBMD has - even without increasing smoothing sensitivity and without local anti-shrinking (orange line) - always a lower mean error than the BMD.

The *lion* and *allegorie* meshes, which correspond to the testdata category of real reconstructions with synthetic noise, are depicted in Figure 4 and 7 respectively. Again, from a visual point of view all three mesh de-



(a) Input (b) BMD (c) Laplacian (d) Our JBMD

Figure 6: Comparison of different mesh denoising algorithms for the *person 1* mesh. Top row: meshes. Bottom row: color-coded comparison against the input mesh.

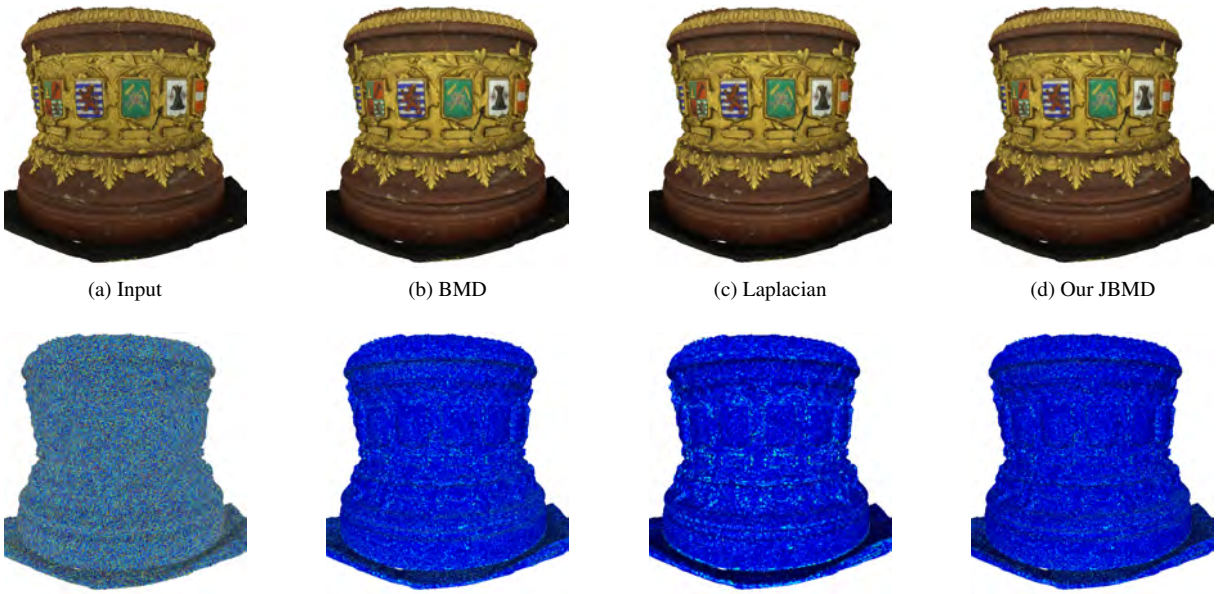
noising algorithms deliver smooth results. However, for both datasets at the edges of the mesh BMD and Laplacian denoising are less precise than our JBMD. Figure 5 and Figure 8 depict the respective mean error of the *lion* and *allegorie* meshes depending on the number of iterations. Again, our JBMD outperforms the competing algorithms and converges at the lowest error level.

The *person 1* and *person 2* meshes correspond to the testdata category of reconstructions with real world noise. Since no groundtruth data is available for these datasets, we compare the denoised meshes against the original mesh in Figure 6 and 9 respectively. The BMD algorithms results in the biggest differences to the original mesh. Especially the nose, but also the eyebrows and mouth, have a huge deviation and are not precise. The Laplacian smoothing shows less deviation, but is by far not as close to the original mesh as our JBMD. Of course, smaller deviations to the original do not mandatory result in a better quality, but from a visual point of view all results are similarly smooth. Thus, also for this category of testdata our JBMD outperforms the competing algorithms.

Summarizing the evaluation results, we found out that our JBMD algorithm outperforms competing algorithms for all tested datasets in terms of precision while creating smooth results. Notably is in particular that our JBMD converges on the lowest error level for higher numbers of iterations. This is the achievement of all three main contributions of our paper: Using color information as additional constraint, correcting shrinking effects locally and increasing the smoothing sensitivity with each iteration. Like illustrated in Figure 3, 5 and 8 this is only possible with the combination of all these three contributions. As long as at least one of them is not activated, the mean error is not minimal and does not converge. With the local anti-shrinking it is possible to denoise especially edges very precisely. Furthermore, we verified with our convincing result that the coherence assumption of coinciding geometry and color changes holds also for meshes.

5 CONCLUSION

In this paper we proposed the new Joint Bilateral Mesh Denoising (JBMD), which is an anisotropic filter for highly precise and smooth mesh denoising. Under the assumption of coinciding geometry and color changes it uses color information as an additional constraint for denoising. This assumption is adapted from the Joint Bilateral Filter (JBF) of the recent image processing research and we showed in this paper that this coherence assumption also holds for meshes. Furthermore, we proposed a precise local anti-shrinking, which leads to precision improvements especially at the edges of the mesh. Our third contribution increases the smoothing sensitivity for higher numbers of iterations. In our evaluation we compared our new JBMD algorithm against competing algorithm based on three categories of test data. We showed that our contributions lead to high precision results with lowest errors. In addition our algorithm converges to the minimum error level for higher numbers of iterations.



(a) Input (b) BMD (c) Laplacian (d) Our JBMD

Figure 7: Comparison of different mesh denoising algorithms for the *allegorie* mesh. Top row: meshes. Bottom row: color-coded error distribution.

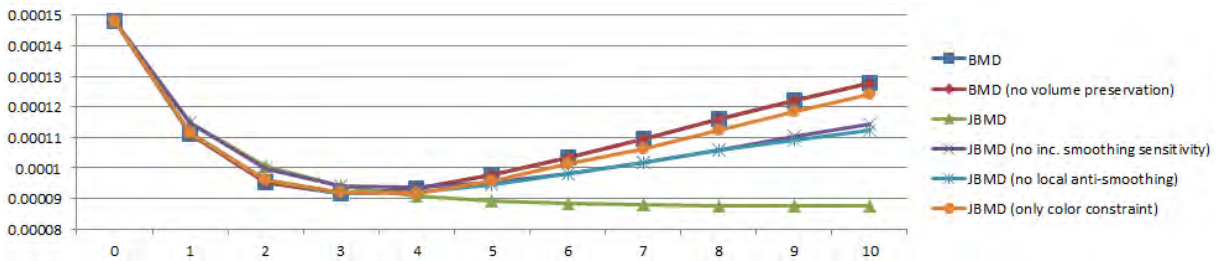
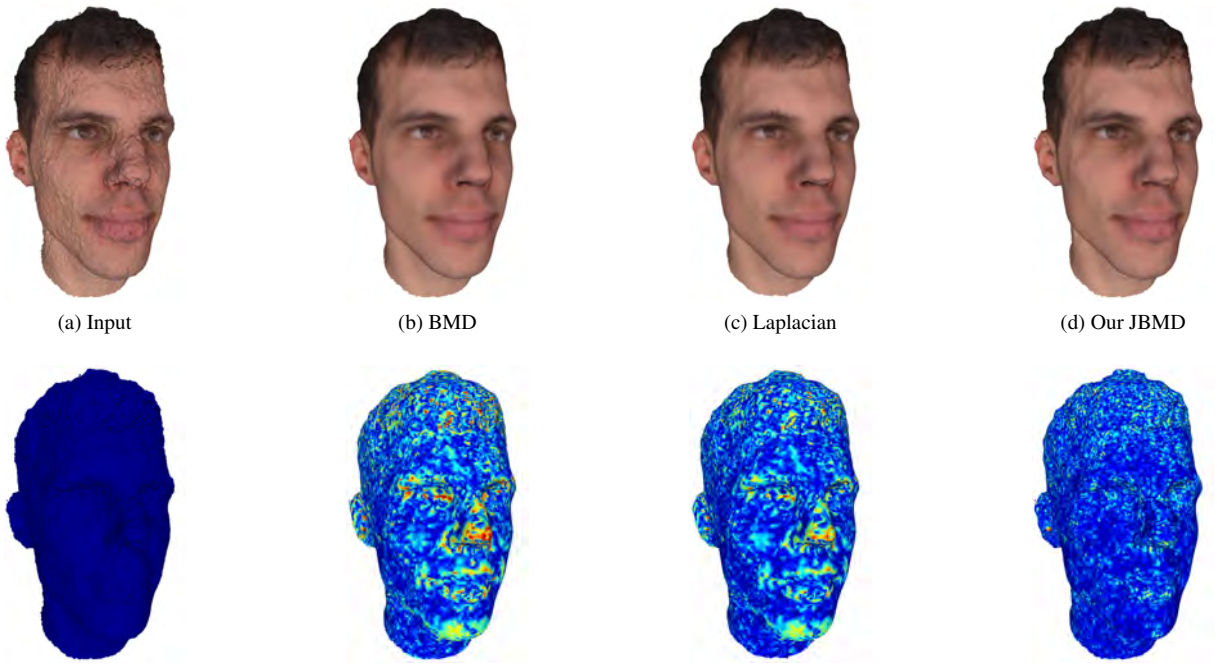


Figure 8: Mean errors of the *allegorie* mesh for different numbers of iterations. Quantitative comparison of different features of our new JBMD algorithm and comparison against BMD.



(a) Input (b) BMD (c) Laplacian (d) Our JBMD

Figure 9: Comparison of different mesh denoising algorithms for the *person 2* mesh. Top row: meshes. Bottom row: color-coded comparison against the input mesh.

ACKNOWLEDGEMENTS

This work was partially funded by the Federal Ministry of Education and Research (Germany) in the context of the project DENSITY (01IW12001).

6 REFERENCES

[Agi] Agisoft. PhotoScan. `http://www.agisoft.com/`.

[BSMH98] Michael J Black, Guillermo Sapiro, David H Marimont, and David Heeger. Robust anisotropic diffusion. *Image Processing, IEEE Transactions on*, 7(3):421–432, 1998.

[DMSB99] Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 317–324. ACM Press/Addison-Wesley Publishing Co., 1999.

[Don95] David L Donoho. De-noising by soft-thresholding. *Information Theory, IEEE Transactions on*, 41(3):613–627, 1995.

[FDCO03] Shachar Fleishman, Iddo Drori, and Daniel Cohen-Or. Bilateral mesh denoising. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 950–953. ACM, 2003.

[Fie88] David A Field. Laplacian smoothing and delaunay triangulations. *Communications in applied numerical methods*, 4(6):709–712, 1988.

[FP10] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(8):1362–1376, 2010.

[HSJS08] Benjamin Huhle, Timo Schairer, Philipp Jenke, and Wolfgang Straßer. Robust non-local denoising of colored depth data. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–7. IEEE, 2008.

[JDD03] Thouis R Jones, Frédo Durand, and Mathieu Desbrun. Non-iterative, feature-preserving mesh smoothing. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 943–949. ACM, 2003.

[KCLU07] Johannes Kopf, Michael F Cohen, Dani Lischinski, and Matt Uyttendaele. Joint bilateral upsampling. In *ACM Transactions on Graphics (TOG)*, volume 26, page 96. ACM, 2007.

[KNRS13] Johannes Kohler, Tobias Noll, Gerd Reis, and Didier Stricker. A full-spherical device for simultaneous geometry and reflectance acquisition. In *Applications of Computer Vision (WACV), 2013 IEEE Workshop on*, pages 355–362. IEEE, 2013.

[NIH+11] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

[PM90] Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(7):629–639, 1990.

[ROF92] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992.

[SSC14] Frank Steinbrucker, Jurgen Sturm, and Daniel Cremers. Volumetric 3d mapping in real-time on a cpu. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2021–2028. IEEE, 2014.

[Tau95] Gabriel Taubin. A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 351–358. ACM, 1995.

[TM98] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846. IEEE, 1998.

[VMM99] Jörg Vollmer, Robert Mencl, and Heinrich Mueller. Improved laplacian smoothing of noisy surface meshes. In *Computer Graphics Forum*, volume 18, pages 131–138. Wiley Online Library, 1999.

[WBS15] Oliver Wasenmüller, Gabriele Bleser, and Didier Stricker. Combined bilateral filter for enhanced real-time upsampling of depth images. *International Conference on Computer Vision Theory and Applications*, 2015.

[ZFAT11] Youyi Zheng, Hongbo Fu, OK-C Au, and Chiew-Lan Tai. Bilateral normal filtering for mesh denoising. *Visualization and Computer Graphics, IEEE Transactions on*, 17(10):1521–1530, 2011.

Using Mutual Independence of Slow Features for Improved Information Extraction and Better Hand-Pose Classification

Aditya Tewari*†
Aditya.Tewari@dfki.de

Bertram Taetz*
Bertram.Taetz@dfki.de
Didier Stricker*
Didier.Stricker@dfki.de

Frédéric Grandidier†
Frederic.Grandidier@iee.lu

ABSTRACT

We propose a Slow Feature Analysis (SFA) based classification of hand-poses and demonstrate that the property of mutual independence of the slow feature functions improves the classification performance. SFA extracts functions that describe trends in a time series data and is capable of isolating noise from information while conserving high-frequency components of the data which are consistently present over time or in the set of data points. SFA is a useful knowledge extraction method that can be modified to identify functions which are well suited for distinguishing classes. We show that by using the orthogonality property of SFA our information about classes can be increased. This is demonstrated by classification results on the well known MNIST dataset for hand written digit detection.

Furthermore, we use a hand-pose dataset with five possible classes to show the performance of SFA. It consistently achieves a detection rate of over 96% for each class. We compare the classification results on shape descriptive physical features, on the Principal Component Analysis (PCA) and the non-linear dimensionality reduction (NLDR) for manifold learning. We show that a simple variance based decision algorithm for SFA gives higher recognition rates than K-Nearest Neighbour (KNN), on physical features, PCA and non-linear low dimensional representation. Finally, we examine Convolutional Neural Networks (CNN) in relation with SFA.

Keywords

Slow Feature Analysis, Hand-Pose Identification, Knowledge extraction, Feature Learning

1 INTRODUCTION AND BACKGROUND

The hand is probably the most effective tool for indicating and gesticulating. Estimating the hand-pose in frames of a sequence to detect a gesture is a common step used in various gesture recognition approaches. Hand gesture recognition is steadily gaining popularity in tasks like navigation, selection and manipulation in Human Computer Interactions [BVBC04]. While complex applications like surgical simulation and training systems require dynamic hand gesture recognition [LTCK03], simpler command and control interfaces often employ hand-poses.

The hand-pose at each frame is treated as a feature in some approaches [CGP07], while some methods use this information to describe the states of a state machine [GMR+02]. A sensor free, vision based detection of pose is a challenging task because of the large degree of freedom in the movement of hand parts and self occlusion that might occur, moreover the calculation of local edge or corner based features is prone to noise [CGP08]. Some methods use physical features of the hand like the gravitational center of the palm region and the finger location [RYZ11]. Other features include convexity that describes the curvature of the palm hull. The works of [PKK09, CLEL12] describe the use of geometrical descriptors for posture detection. We argue that because of occlusion and the high degree of freedom, high level features learnt from hand-pose data can help in improving the classification. In [LCP12] a method of manifold embedding for articulated hand configuration detection is proposed. This method learns one of the global description of data by identifying the manifold on which the data resides.

The SFA allows unsupervised learning of invariant or slowly varying features. It can learn translation, scale and rotational invariances [WS02]. The SFA technique has been modified to achieve supervised learning to

* DFKI, Augmented Vision, Trippstadter Str.122, Germany, 67663 Kaiserslautern.
† IEE-SA, Weiergewan, 11-rue Edmond Reuter, Luxembourg, 5326 Contern.

achieve classification [Ber05]. It provides mutually orthogonal features thus the prominent features carry independent information about the data even though they remain invariant to size, rotation and translation. Another important property of the SFA is the guaranteed optimisation to the slowest changing function which allows for easy extension when learning a new class. We propose to learn several slow feature functions for each class to improve classification further. To achieve this we employ the property of mutual orthogonality of features learnt from a class. The mutual orthogonality of SFA features result in aggregation of information thus it increases the effective information that a classifier receives.

Section 2.1 describes the basic ideas behind SFA, further we discuss its use as a classifier in section 2.2. In section 2.3 we explain the use of orthogonality to increase information and describe its effect on the classification task on the MNIST dataset in section 3.1. Finally in section 3.2 we apply the technique on hand-pose classification.

We ascertain the applicability of SFA as an information extraction method, by demonstrating better classification rates as compared to the standard PCA and manifold learning methods. The manifolds representation of data compensates for non-linearities. The better performance of SFA over manifold learning proves its strong capability of identifying the consistent properties of signals in a dataset. Apart from the comparison with PCA and manifold learning methods we also make comparisons to classification performed by using shape descriptors and geometrical features calculated from the hand-pose images. We report a substantial improvement over the classification done with these features. The improvement over physical features indicates that SFA is capable of information extraction while the improved classification compared to manifold embedding establishes the ability to handle non-linearities in a dataset.

The broad contributions made through this work are:

- The applicability of SFA for hand-pose classification using data obtained from a time of flight camera.
- SFA classification based on several slow feature functions and not just the principal slow feature.
- A comparison of classification based on physical features and SFA features that indicates the superior information extraction capability of SFA.
- The demonstration of improved classification performance on the MNIST hand written digit dataset and the hand-pose dataset using a modified SFA.

2 SLOW FEATURE ANALYSIS

2.1 Slow Feature Analysis as a Learning Problem

Low level features are short duration features and are often misleading. High level features of the data carry information that extends beyond small neighbourhoods. SFA learns functions that represent such high level features. These high level representation can better explain the property of the data space. A feature that does not vary rapidly, yet has a slow consistent change promises to describe the behaviour of a function in better detail [Föl91]. The slow features thus provide a consistent trend in the data. The SFA is originally designed for detection of trends in temporal data [WS02]. It has been modified to provide consistent trends within elements belonging to a static dataset [Ber05]. We first discuss the SFA procedure for temporal data and shall later explain the modifications for classification in static datasets.

If a vectorial input $\mathbf{X}(\mathbf{t}) \in \mathbb{R}^d$ is a time series, one of the slow features is the function $\mathbf{g}(\cdot)$, such that $\mathbf{y}(\mathbf{t}) = \mathbf{g}(\mathbf{X}(\mathbf{t}))$, varies as slowly as possible while avoiding trivial responses.

The problem is formally described by [Wis03] as minimising the absolute differential

$$\Delta(y_j) := \langle \dot{y_j}^2 \rangle. \tag{1}$$

Here y_j is the jth component of $\mathbf{y}(\mathbf{t})$ and $\dot{y_j}$ is the derivative of y_j with respect to time t and $\langle \cdot \rangle$ denotes average over time. The absolute differential is minimised under the following conditions:

$$\langle y_j \rangle = 0 \tag{2}$$

$$\langle y_j^2 \rangle = 1 \tag{3}$$

$$\langle y_i y_j \rangle = 0 \quad i \neq j. \tag{4}$$

While the minimisation selects invariant features, (3) forces some variance and removes the possibility of obsolete solutions like a constant function and (4) forces independence among the calculated slow features. These constraints are forced by sphering the data [LZ98].

Sphering of $\mathbf{X} \in \mathbb{R}^d$ means we transform $\mathbf{X}$ such that the covariance matrix of the transformed random variable $\mathbf{X}^*(\mathbf{t})$ is an identity matrix. $\mathbf{X} = (x_1, x_2 ..., x_n)$, represents a data matrix and $x_1, x_2, x_3, ..., x_n$ are n vectors belonging to it. If $(\mathbf{X} - \mu)$ and Σ are respectively the centered data matrix and the covariance matrix, then the sphered data is expressed as:

$$\mathbf{X}^*(\mathbf{t}) = B_n(\mathbf{X} - \mu), \quad with \quad B_n^T B_n = \Sigma^{-1}. \tag{5}$$

The sphered data $\mathbf{X}^*(\mathbf{t})$ is projected into a quadratic space, resulting in data $\mathbf{Z}$. The derivative $Z(t+1) - Z(t)$, is represented by $\dot{Z}$. Let $\mathbf{W}$ be the

eigenvectors of the covariance matrix of the derivative matrix $\dot{\mathbf{Z}}$,

$$\langle \dot{\mathbf{Z}}\dot{\mathbf{Z}}^{\mathbf{T}} \rangle \mathbf{W} = \lambda_{\mathbf{W}}. \quad (6)$$

The eigenvectors corresponding to the smallest eigenvalues are the direction of the slowest change in differential of the data. These eigenvectors compose the slow feature functions. These functions are the weighted linear sums over the components of the expanded signal, where weights are the components of eigenvectors w,

$$g_j(x) = {w_j}^T.Z(t). \quad (7)$$

Where w_j is the jth column of the matrix $\mathbf{W}$. The m smallest eigenvalues correspond to the m primary slow feature functions: $g_1, g_2, g_3 \cdots g_m$.

2.2 Slow Feature Analysis for Classification

The slow features describe intrinsic features of a long time series. It is the property of slow features to conserve variations over time, this property can be exploited for classification. The data for classification is not temporal and thus the absolute differential described in (1) is modified to perform a supervised classification. To perform a supervised classification, functions resulting in minimum inter-element difference within each class are identified. As in case of time series SFA, the conditions of zero mean, constant variance and linear independence are imposed. Once again these conditions are satisfied by sphering the data. Furthermore, the optimisation process tries to increase the variance outside a class, to identify the slow feature functions.

For the dataset $\mathbf{X}$, we define a matrix $\mathbf{Z}$, such that $\mathbf{Z}$ is the quadratic expansion of the sphered transform of $\mathbf{X}$. Accordingly, the differential term for a vector z^{el} belonging to the expanded dataset $\mathbf{Z}$ is represented as:

$$\nabla_{el} := \sum_{C=1}^{N} \sqrt{\sum_{n=1}^{N_C} (z_C^n - z^{el})^2}. \quad (8)$$

Thus average differential for the data $\mathbf{Z}$ can be re-represented as:

$$\nabla := \langle \nabla_{el} \rangle. \quad (9)$$

Where, z^{el} is the vector corresponding to the element for which the differential is calculated. z_C^n is the nth element of a class C, N is the number of classes and N_C is the number of datapoints in the class C. We now minimise the value of ∇. This minimisation condition returns functions that forces slow variance within classes. Each of the slow features correspond to one of the classes, to further improve the extracted feature functions, (9) is extended to maximise the variance between classes while minimising it within the class [ZT12].

To achieve this we subtract the average of the absolute difference of the in-class element with elements outside the class (∇_{el}^o) from the average differential within the class (∇_{el}), that yields

$$\nabla_{el}^o := \sum_{C=1}^{N} \sqrt{\sum_{\{c=1, c\neq C\}}^{N} \sum_{n=1}^{N_c} (z_c^n - z^{el})^2}. \quad (10)$$

The calculation of the slow feature function is modified to minimising the cost function O, where O is defined as:

$$O = \langle \nabla_{el} \rangle - \langle \nabla_{el}^o \rangle. \quad (11)$$

2.3 Using Orthogonality to Increase Information

The classification process described above returns (N=number of classes) functions. These functions are learnt from the entire dataset using the optimisation function of (11). This procedure results in a set of functions which provide low variance response. The constraint of decorrelation between different slow features creates the possibility of learning many functions corresponding to one class.

The ready availability of features after doing an SFA procedure, and there mutual independence motivates us to find more features within a class. Thus we calculate multiple slow features corresponding to each class. Rather than learning slow features over the entire dataset we learn a set of function for every class. Slow features are learnt by restricting the dataset to elements of one class, this is repeated for all classes.

As each function is orthogonal, we have more than one function representing intrinsic properties of the specific class. These linear functions are decorrelated on the expanded space. Learning slow features in every class requires a larger training dataset, meanwhile it also results in adding information for classification. The optimisation function (11) is further modified to minimise variance within a class, while maximising out-of-class variation using all other classes (13). This modification extends (10) as follows:

$$\nabla_{el_C} := \sqrt{\sum_{n=1}^{N_C} (z_C^n - z^{el_C})^2}, \quad (12)$$

$$\nabla_{el_C}^o := \sqrt{\sum_{\{c=1, c\neq C\}}^{N} \sum_{n=1}^{N_c} (z_c^n - z^{el_C})^2}, \quad (13)$$

$\nabla_{el_C}^o$ in (13) is the sum of out-of-class variances calculated over the training dataset.

$$O_C = \langle \nabla_{el_C} \rangle - \langle \nabla^o_{el_C} \rangle. \quad (14)$$

el_C represents that the calculation for the differential is done for elements belonging to the class C. The optimisation for class C is achieved by minimising O_C.
The functions are collected as matrix $\mathbf{W_C}$ where C is the class for which these functions are learnt. $w_{\lambda_{C_j}}$ is the vector corresponding to the jth eigenvalue λ_{C_j} of class matrix $\mathbf{W_C}$. For a test input vector $\mathbf{P}$ the functional $\mathbf{G}$ returns an output vector $\mathbf{G}(\mathbf{W},\mathbf{P})$. The functional $\mathbf{G}$ has m linear functions in the space corresponding to the dimension of vector expanded in data space,

$$G(\mathbf{W_C},\mathbf{P}) = \mathbf{P} \cdot \bar{\mathbf{W}}_{\mathbf{c}}{}^T. \quad (15)$$

The variance for the output of the function is calculated as,

$$Var_C \quad = \sum_j (\mathbf{P} \cdot \bar{w}_{\lambda_{C_j}})^2 \quad = \sum G(\mathbf{W_C},\mathbf{P})^2. \quad (16)$$

The final classification is performed as follows:

$$class = \underset{C}{\operatorname{argmin}}(Var_C). \quad (17)$$

While doing an N class classification using m functions for each class, we have Nm functions. Some of these functions are very similar even though they belong to separate classes. This does not affect the minimum variance choice, because of aggregation.
The value of functions corresponding to a class when applied to an element from the same class is centred around a constant value. When a function is applied on a mismatched class, the result is random. This randomness likely results in a wrong identification.
In the case of multiple centred functions, corresponding to a class, the resulting output for a matching sample has all the function outputs centred around zero. Some functions from non-matching classes may return centred responses close to zero but, the aggregated variance for a mismatch element is higher, resulting in clearer distinction from the matching class.

3 EXPERIMENTS

3.1 Effect of Increased Information on MNIST Dataset

MNIST dataset [LC12] is one of the most popular dataset for evaluating classification problems. The Lecun network [LJB+95] has achieved an error rate of less than 0.3% on the MNIST dataset. [Ber05] also describes the original classification technique on the MNIST Hand written digit dataset. We further tested and compared both methods of using SFA for classification described earlier on the same dataset. Each datapoint in the MNIST dataset is a 28x28 pixel image. We reduce it to a 35 dimensional vector by employing PCA and then project it into a quadratic space. The quadratic expansion of the 35 dimensional PCA vector results in a vector of size 630. We calculate 10 slow features functions for the full dataset. Also, we calculate 10 slow feature functions for each class. It was observed that the identification performance for every class improved when we used the property of orthogonality to calculate slow feature functions. The comparative results are listed in Table 1.

class	0	1	2	3	4	5	6	7	8	9
Full Dataset	81	93	79	83	77	72	77	80	73	84
Class Separation	91	96	82	85	79	81	89	91	83	84

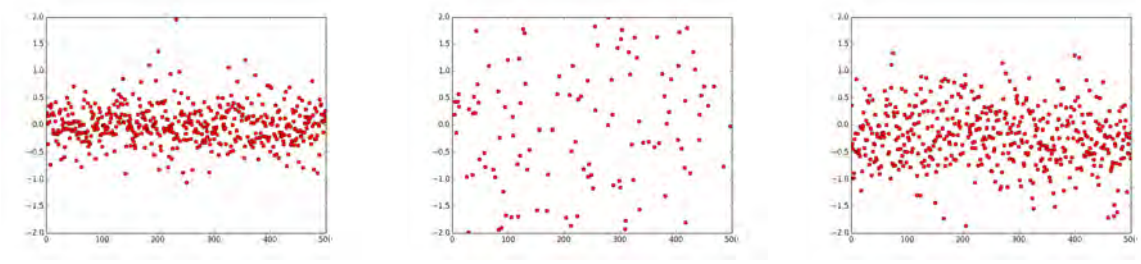
Table 1: Classification accuracy in % for each digit mentioned on the top row. The second row values are accuracy percentages when slow feature functions are learnt from the entire dataset, the third row shows the accuracy percentages when several functions are learnt independently for each class

Figure 1 and Figure 2 show the difference between the two methods for classification. Figure 1 is based on identification of feature function from the entire data while Figure 2 is based on the classification approach where multiple corresponding functions are learnt from each class. The Y axis represents the distance of the response from the mean response calculated during the training stage, the X axis marks the index of input element on the dataset. The input elements are stacked in order of the classes that they belong to.
Figure 1 shows the centred response of the first three classes to the function corresponding to class with digit 0. The deviation of elements of class '0' from the origin are smaller as compared to other classes. This fits our hypothesis that SFA looks for feature functions that minimise the in-class variance. The Figure 2 shows the response of each data point to three functions learnt for class 0. The response of the data points of each class is shown in the same figure, with dark blue (the first cluster) representing class 0. The lower variance of function value to the matching class is clearly visible in these figures, the aggregation of function 1, 2 and 3 results in a deviation which is smaller for the matching class, but higher for mismatch. Averaging over these function values reduces the likely possibility of error in the first method because of randomness of non matching function response.

3.2 Hand-pose Experiments

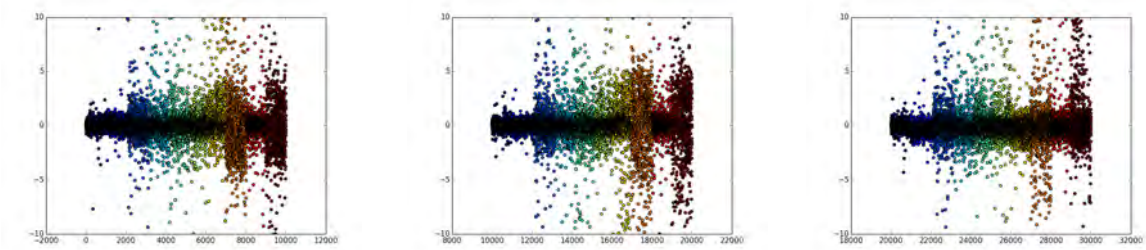
3.2.1 Hand-Pose Data Collection

A 3D Time-of-light, PMD-Nano camera has been used to collect a dataset of hand-poses. The camera is fixed vertically above the palm. The output of the PMD-Nano time of flight camera is an 120x165x2 image. The two channels of the image are the amplitude



(a) class:0 (b) class:1 (c) class:2

Figure 1: Response of class 0, 1 and 2 to function learnt from class 0.



(a) function:1 (b) function:2 (c) function:3

Figure 2: Response of all classes to the first 3 functions learnt from class 0.

value and the depth map image data. We cover the arm region with absorbent clothing and use the reflectance of skin to identify the palm. The reflectance constraint does not entirely remove the background and thus the closest contour greater than a threshold area is chosen as the palm region. The segmented palm region is then converted into a binary image which is further used for hand-pose identification.

We then learn slow feature functions for five hand-pose classes labelled as "Fist", "Flat", "Index", "Open" and "Grab", see Figure 3. Slow features or invariances are learnt from a dataset of 3,000 frames of each class from 3 subjects. 1000 frames in each class are randomly selected and rotated in either direction, by an angle between 10° and 20°. These rotated frames are added to the training dataset along with the original frames. Note that, this spreads the poses such that they cover the whole rotational axis, it also increases the dataset and generates samples which train the SFA for rotational invariances.

Three hundred frames are selected for each class through random partitioning of the original dataset. These samples are used as test dataset, while the remaining original dataset is used for training. The preprocessing follows the same procedure as described for the training dataset.

3.2.2 *Hand-pose Identification*

Before learning slow features from the dataset of segmented hands, the image is scaled down to one-third of its original size. This is followed by a PCA which reduces each image to a 35 dimension vector that is projected to its quadratic space to allow the learning of non-linear invariances in the principal components of the training data.

During the SFA learning process the covariance matrix

(a) Fist (b) Flat (c) index (d) Open (e) Grab

Figure 3: The hand-pose samples.

of the differential data as well as the eigenvectors corresponding to the largest eigenvalues are recorded. The eigenvectors corresponding to the ten largest eigenvalues correspond to the linear functions used for classification. Each function is centred around the mean values learnt during the training process. It is observed that the samples of matching classes are tightly spread around the mean values of the classes. The class which corresponds to the function has much smaller variance as compared to other classes. Figure 4 shows the response of the test dataset on the most prominent function of the "Fist" class. The data points for each class are represented by a unique color. The "Fist" class which is represented by blue in the figure has relatively tight packing of the data-points as compared to any other class. Like in the previous figures the X axis of the plot represents the data points which are arranged by their labels, and the Y axis represents the centered value of the learnt function.

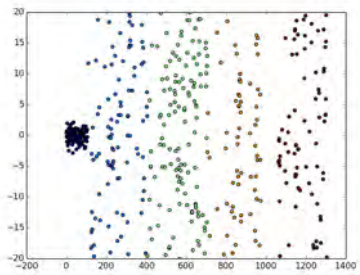


Figure 4: Slow function response for class 'Fist'.

This pattern is visible over the entire set of *Nm* functions, 5x10=50 in the present case. Table 2 shows the response of each data-point to the first five of the 10 learnt functions of each class. Each rows represents a set of functions corresponding to the class.

It can be observed that the functions learnt for one class have lower variance in the same class, while higher variance in other classes. This observation is used to differentiate classes. Thus we calculate the variance of the function response over all the functions calculated for a class.

The three hundred frames of each class in the dataset are used for evaluations. While learning models that are saved include, PCA mapping for each class, the sphering matrix, *m* eigenvectors and the covariance matrices for each class.

4 RESULTS AND COMPARISONS

We compare the results of the classification using slow feature analysis with results from KNN on physical features extracted from each frame. The physical features include coordinates of the tip of the finger (or the tip of the palm), the coordinates of the palm centroid, the convex ratio and the concave depth of the image and the polar and azimuth angle of the finger [RYZ11, PKK09, CLEL12] . We also compare the results to KNN applied on the PCA of the data and the low dimension manifold of the raw binary image [LCP12].

The KNN models for the physical features are generated using 1500 samples from each class and are modelled by simple euclidean distances. The Manifold is learned by Isomap algorithm [TDSL00] and the learning is done by the same training data as used for slow feature analysis.

Slow Feature Analysis based classification works better than the physical feature based classification evaluated in the KNN model. It also outperforms the KNN evaluation done with 35-dimensional (35-D) PCA and 9-dimensional (9-D) manifold representation of the dataset. We chose 35-D PCA because it is used as the basis for SFA calculation and 9-D isomap because the classification by KNN performs best for it. Table 3 shows the confusion matrix for the SFA based classification, Table 4 shows the confusion matrix for classification on KNN model trained on the hand crafted physical features. Table 5 is the confusion matrix for classification results from KNN model trained on the 35-D PCA representation of the image data. While classifying on the 9-D element vector received from the isomap done on the palm region as described earlier, the results are improved as compared to KNN on physical features and PCA based KNN. Table 6.

The results from the SFA are considerably better than the results from the physical features. These features are carefully selected for hand-pose estimation. This underlines the ability of the method to search for relevant features in a class. This improvement also suggests that SFA is capable of reducing the effect of local noise and distortion.

We compare SFA with KNN on the lower dimension representation of the data computed by PCA. The confusion matrices of Tables 3 and 5 clearly demonstrate that SFA performs far better. Thus the process of calculating the slow feature functions after doing PCA on the data further refines the knowledge that we are able to extract from the dataset.

SFA classification also performs better than a KNN model trained on manifold representation of the dataset. While the identification of the "Flat" hand-pose is better than the SFA in case of the isomap representation, the overall performance of SFA is superior. It is notable that KNN is a far more complex model as compared to simpler variance based classification of SFA. This result suggests that SFA is capable of managing non-linearities in the data, this can be attributed to the step in which the PCA data is projected onto a quadratic space.

The improvement from PCA to isomap modelling is a result of better handling of non-linearities in the data. The KNN model based on euclidean distances suffers from the inability to compensate for non-linearities, this is overcome when we use the isomap projection. It is also important to note that while the KNN model is learnt over the isomap projection, SFA classification provides better results by simple variance calculations. It is worth mentioning that the performance improvement in the quality of classification was minimal when we scaled the palm region by distances. This observation can be attributed to the characteristic of SFA that, it explores multidimensional linear functions which encompasses the invariances over the data points.

Discussion

The SFA, as demonstrated in the last section, performs well for the classification task. Even though the total labelled data available to us was small, we compared the performance of SFA classification for hand-pose with CNN. The CNNs have resulted in exceptional classification results. As mentioned earlier Lecun network

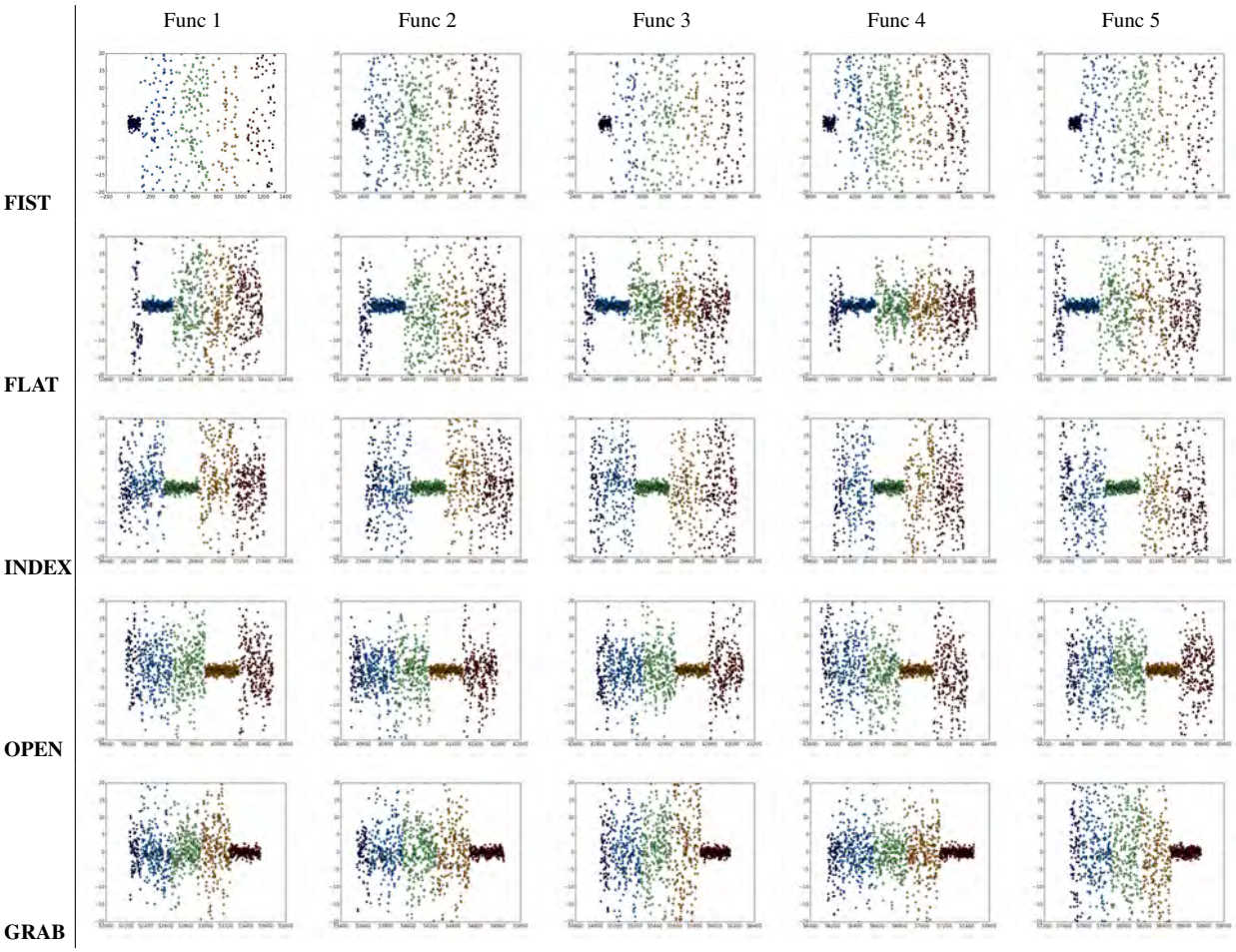


Table 2: Scatter map showing the value for 5 SFA functions for every class on the test dataset, different colors represent different classes.

based on CNN has achieved an error rate of less than 0.3% on the MNIST dataset, this compares favourably with human accuracy. We tested our hand-pose dataset for training a CNN with two convolution layers and two Max pooling layers. Using 15000 data-points after rescaling. The accuracy of classification reached over 98% after 30,000 iteration with a batch size of 50 images. Although, it was observed that because of the relatively small amount of data the CNN model starts over-fitting. The use of easily available, less specific hand-pose datasets for pre-training the CNN is one of the possible methods of overcoming the problem of over-fitting with the present data. SFA also requires a large dataset but lesser than CNN, we demonstrate that it is capable of learning functions for each class of hand-poses with 3000 data points. It can be argued that the SFA learning process results in learning of information that defines the class of the dataset, but the convolutional features learnt by a CNN using the classification based method contain information that distinguishes different classes. SFA results in lesser classification accuracy than CNN on a large dataset, but SFA gives interpretation about the nature of the class independent, which seems to be harder to identify in a CNN model.

5 CONCLUSION

In this paper we used SFA for classification on two datasets. SFA was tested on MNIST dataset and a hand-pose dataset. We approached the classification by training SFA separately for each class and demonstrated that, the property of orthogonality of SFA helps in extracting more information about the class. We showed that SFA outperforms hand picked physical features for hand-pose classification. This confirms the recent trend of preferring global features which are learnt from the data over extracting features by intuition. Training and test data has considerable variances of rotation and scale, in our experiments SFA remains robust to such variances.

The use of slow feature analysis also reduces the on line processing required on the test sample. SFA based classification requires a relatively large dataset for training, additionally it employs an expensive batch learning algorithm which requires large computer memory to run. Yet, it displays a remarkable ability to extract information and identify trends in a dataset. Usually calculating

%	FIST	FLAT	INDEX	OPEN	GRAB
FIST	**97.0**	1.0	0.0	1.7	0.3
FLAT	0.0	**96.7**	2.3	1.0	0.0
INDEX	0.0	0.0	**98.7**	1.3	0.0
OPEN	1.0	0.0	1.3	**97.6**	0.0
GRAB	0.7	2.3	0	0.3	**96.7**

Table 3: Confusion matrix for SFA classification. Bold values are accuracy values for the class corresponding to the respective row.

%	FIST	FLAT	INDEX	OPEN	GRAB
FIST	**97.0**	0.7	1.3	0	1.0
FLAT	0.7	**95.7**	3.0	0	0.7
INDEX	2.7	5.7	**91.7**	0.3	0.0
OPEN	3.0	2.3	0	**94.3**	0.3
GRAB	0.7	4.7	0	0.3	**94.3**

Table 4: Confusion Matrix for KNN classification based on physical features. Bold values are accuracy values for class corresponding to the respective row.

%	FIST	FLAT	INDEX	OPEN	GRAB
FIST	**78.3**	12.2	2.9	3.8	2.9
FLAT	1.3	**80.7**	6.3	6.6	5.0
INDEX	0.0	3.3	**81.7**	2.0	14.0
OPEN	0.0	7.7	4.7	**85.3**	2.3
GRAB	0.3	3.3	7.7	3.0	**85.7**

Table 5: Confusion Matrix for KNN classification results on 35-D PCA. Bold values are accuracy values for the class corresponding to the respective row.

%	FIST	FLAT	INDEX	OPEN	GRAB
FIST	**97.0**	0.3	2.0	0.7	0
FLAT	0.3	**98.3**	1.3	0	0.3
INDEX	3.7	0.3	**96.0**	0	0
OPEN	1.7	0.0	1.0	**96.3**	1.0
GRAB	2.7	0.3	0.3	0.7	**96.0**

Table 6: Confusion Matrix for KNN classification on 9-D isomap on raw images. Bold values are accuracy values for the class corresponding to the respective row.

features at run time is a hard task, it consumes considerable computing and development effort. Whereas, SFA requires few linear operations to calculate the slow features. Thus, it does not only improve the robustness towards the data but also improves the performance of the machine when compared with processes that use physical features.

We showed the performance on global SFA features in this work and compared it to physical (local) features. Note that when we tested the SFA for classification of fixed length time series sequences, local features like peaks and inflexion, when combined with slow features, improved the classification performance. Classification was made using a logistic regression classifier. However, this fusion requires online feature calculation and a more complex classifier model.

It will be interesting to further study and quantify the effect of noise and poor segmentation on these features. Also further experiments with various data sources and the influence of an increasing number of classes on the orthogonality property of SFA will be of interest. We plan to extend the present approach of pose detection to gesture recognition. The batch learning approach is not suitable for the gesture classification and recently developed incremental SFA [KLS11] is a promising solution to the problem.

6 ACKNOWLEDGMENT

This work is supported by the National Research Fund, Luxembourg, under the AFR project 7019190.

7 REFERENCES

[Ber05] Pietro Berkes. Pattern recognition with slow feature analysis, February 2005.

[BVBC04] Volkert Buchmann, Stephen Violich, Mark Billinghurst, and Andy Cockburn. Fingartips: gesture based direct manipulation in augmented reality. In *Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 212–221. ACM, 2004.

[CGP07] Qing Chen, Nicolas D Georganas, and Emil M Petriu. Real-time vision-based hand gesture recognition using haar-like features. In *Instrumentation and Measurement Technology Conference Proceedings, 2007. IMTC 2007. IEEE*, pages 1–6. IEEE, 2007.

[CGP08] Qing Chen, Nicolas D Georganas, and Emil M Petriu. Hand gesture recognition using haar-like features and a stochastic context-free grammar. *Instrumentation and Measurement, IEEE Transactions on*, 57(8):1562–1571, 2008.

[CLEL12] J-F Collumeau, Rémy Leconge, Bruno Emile, and Hélène Laurent. Hand gesture recognition using a dedicated geometric descriptor. In *Image Processing Theory, Tools and Applications (IPTA), 2012 3rd International Conference on*, pages 287–292. IEEE, 2012.

[Föl91] Peter Földiák. Learning invariance from transformation sequences. *Neural Computation*, 3(2):194–200, 1991.

[GMR+02] Namita Gupta, Pooja Mittal, S Dutta Roy, Santanu Chaudhury, and Subhashis Banerjee. Developing a gesture-based interface. *Journal of the Institution of Electronics and Telecommunication Engineers*, 48(3):237–244, 2002.

[KLS11] Varun Raj Kompella, Matthew D Luciw, and Jürgen Schmidhuber. Incremental slow feature analysis. In *IJCAI*, volume 11, pages 1354–1359, 2011.

[LC12] Yann LeCun and Corinna Cortes. The mnist database of handwritten digits, 1998, 2012.

[LCP12] Chan-Su Lee, Sung Yong Chun, and Shin Won Park. Articulated hand configuration and rotation estimation using extended torus manifold embedding. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 441–444. IEEE, 2012.

[LJB+95] Yann LeCun, LD Jackel, Léon Bottou, Corinna Cortes, John S Denker, Harris Drucker, Isabelle Guyon, UA Muller, E Sackinger, Patrice Simard, et al. Learning algorithms for classification: A comparison on handwritten digit recognition. *Neural networks: the statistical mechanics perspective*, 261:276, 1995.

[LTCK03] Alan Liu, Frank Tendick, Kevin Cleary, and Christoph Kaufmann. A survey of surgical simulation: applications, technology, and education. *Presence: Teleoperators and Virtual Environments*, 12(6):599–614, 2003.

[LZ98] Guoying Li and Jian Zhang. Sphering and its properties. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 119–133, 1998.

[PKK09] Giorgio Panin, Sebastian Klose, and Alois Knoll. Real-time articulated hand detection and pose estimation. In *Advances in Visual Computing*, pages 1131–1140. Springer, 2009.

[RYZ11] Zhou Ren, Junsong Yuan, and Zhengyou Zhang. Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera. In *Proceedings of the 19th ACM International Conference on Multimedia*, MM '11, pages 1093–1096, New York, NY, USA, 2011. ACM.

[TDSL00] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[Wis03] Laurenz Wiskott. Slow feature analysis: A theoretical analysis of optimal free responses. *Neural Computation*, 15(9):2147–2177, 2003.

[WS02] Laurenz Wiskott and Terrence Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural computation*, 14(4):715–770, 2002.

[ZT12] Zhang Zhang and Dacheng Tao. Slow feature analysis for human action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(3):436–450, 2012.

Localized Search for High Definition Video Completion

Jocelyn Benoit
École des arts numériques, de l'animation et du design
Montreal, Canada
jbenoit@nad.ca

Eric Paquette
Multimedia Lab, École de technologie supérieure
Montreal, Canada
eric.paquette@etsmtl.ca

ABSTRACT

This paper presents a new approach for video completion of high-resolution video sequences. Current state-of-the-art exemplar-based methods that use non-parametric patch sampling work well and provide good results for low-resolution video sequences. Unfortunately, because of memory consumption problems and long computation times, these methods handle only relatively low-resolution video sequences. This paper presents a video completion method that can handle much higher resolutions than previous ones. First, to address the problem of long computation times, a dual inpainting-sampling filling-order completion method is proposed. The quality of our results is then significantly improved by a second innovation introducing a coherence-based matches refinement that conducts intelligent and localized searches without relying on approximate searches or compressed data. Finally, with respect to the computation times and memory problems that prevent high-resolution video completion, the third innovation is a new localized search completion approach, which also uses uncompressed data and an exact search. Combined together, these three innovations make it possible to complete high-resolution video sequences, thus leading to a significant increase in resolution as compared to previous works.

Keywords

Video completion, high-resolution, object removal, patches coherence, localized search, multi-resolution

1 INTRODUCTION

Both image and video completion are important tasks in many multimedia applications. Their goal is to automatically fill missing regions of an image/video in a visually plausible manner. Two key factors differentiate video completion from image completion. Firstly, for video completion, it is important to maintain temporal consistency since human vision is more sensitive to temporal artifacts than to spatial artifacts. Using an image completion technique individually on each frame produces undesired temporal artifacts. Secondly, it is more important for video completion to be time- and memory-efficient since video contains much more data than image.

In the past years, many new solutions have been proposed for video completion. It has been shown that exemplar-based methods, that use *non-parametric patch sampling*, work well and provide good results. Unfortunately, they work only on relatively low-resolution videos because larger ones require too much memory. Few methods [8, 12] present results for 640×480 or 540×432 resolutions, with most [6, 7, 9–11, 15, 17, 20, 21] presenting results of 320×240 or lower resolutions. Since High Definition (HD) videos with 1920×1080 or higher resolutions are now commonplace, most of these methods cannot be applied directly or they require too long computation times.

To understand the proposed method, we must first look at the non-parametric patch sampling approaches. Those methods are based on an iteration through each of the patches in the missing regions and a search in all of the patches of the existing regions to find the most similar patch. Without optimization, this search can be excessively time consuming: $O(m^3M^2F)$ with M representing the video width and height; m the patch width, height and depth; and F the number of frames. Even with optimization methods, the search time of the non-parametric patch sampling approaches still remains excessive. Furthermore, the structures needed for theses optimization methods require too much memory, making them inappropriate for HD videos.

Rather than focusing on the acceleration of the nearest neighbors search, the proposed method narrows the search space at finer (higher) resolutions using information obtained at coarser (lower) resolutions. First, let us consider two patches at coarser resolutions: patch w_p^l from the missing region and its most similar patch $w_{p'}^l$ from the existing region. The most sim-

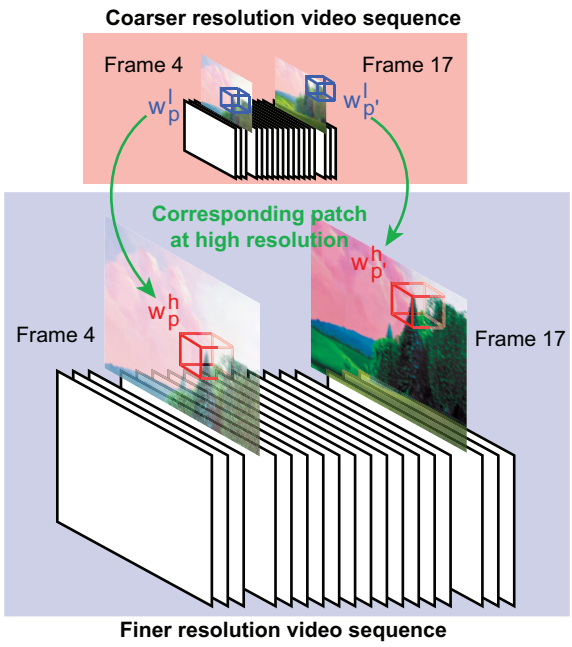


Figure 1: First row: coarser resolution video. Patch (w_p^l) in the missing region and its most similar patch ($w_{p'}^l$) in the existing region. Second row: finer resolution video. The corresponding patch (w_p^h) of w_p^l and its most similar patch ($w_{p'}^h$) in the existing region.

ilar patch of the corresponding patch w_p^h at finer resolutions is likely to be found near $w_{p'}^h$, as illustrated in Figure 1. The proposed approach begins by completing the video at coarser resolutions using a dual inpainting-sampling filling-order completion approach based on Wexler et al. [20]. Since efficient but approximate search approaches are used to find the most similar patches, errors are introduced and several matches are sub-optimal patches. To solve this problem, a coherence-based matches refinement process is used to search for better matches. The technique then stores the space-time location of the most similar patch found for each patch of the missing region in a matches list *ML*. This *ML* is then used by a localized search completion approach to narrow the search space in higher resolution, thus enabling the completion of HD video sequences.

The contributions of the proposed method include a dual inpainting-sampling filling-order completion approach based on Wexler et al. [20]; a new coherence-based matches refinement process that improves the quality of the matches when approximate search approaches are used; and a new localized search completion approach based on an exact search using uncompressed data but restricted to a localized region. We show that the proposed methods enable the completion of HD video sequences and that they produce visually plausible results within reasonable timeframes. Moreover, the approach requires very little memory at the finest resolution except for the input video storage.

2 PREVIOUS WORKS

In past years, many methods have been proposed to replace missing regions of an image. *Image inpainting* techniques propose to fill the missing region by extending the surrounding existing region until the hole vanishes. These techniques generally work only on small and thin holes. *Image completion* techniques use non-parametric patch sampling and are able to fill even larger missing regions of an image. While video completion methods are based on image completion and inpainting methods, video completion poses the additional challenge of maintaining spatio-temporal consistency. Using image completion or image inpainting methods on each frame independently produces temporal artifacts that are easily noticed by the viewer [3].

2.1 Video completion

Extending the image completion methods based on Markov Random Fields (MRF) and non-parametric patch sampling, Wexler et al. [19, 20] address the problem of video completion as a global optimization, and thus obtain good results on relatively large missing regions. Shiratori et al. [17] proposed a similar approach, but find patches based on motion fields instead of color values. Xiao et al. [21] extend these works by formulating video completion as a new global optimization problem defined over a 3D graph defined in the space-time volume of the video. Liu et al. [10] later have proposed an algorithm with two stages: motion fields completion and color completion via global optimization. The major drawback of all these approaches is the amount of information that must be processed when considering HD video sequences. While some methods use per-pixel searches [19–21], other approaches use larger primitives instead of pixels: Shih et al. [16] use fragments, while Cheung et al. [4] use “epitomes”. Approaches using fragments or epitomes can reduce the search time and improve overall coherence, but per-pixel searches are more likely to correctly restore the fine and subtle details found in HD video sequences.

Many methods segment the video sequence into foreground and background parts [6–8, 11, 14] or into layers [22]. These methods create a static background mosaic of the entire sequence, and as a result, these techniques are limited to video sequences with a static background using a fixed camera. Patwardhan et al. [12] later proposed a framework for dealing with videos containing simple camera motions, such as small parallax and hand-held camera motions. The major drawback with all these techniques is that the pixels replaced are static across the video sequence, thus removing details such as video noise, film grain, or slightly moving objects, such as tree leaves, from the background. At an

HD resolution, this lack of detail is quickly noticed by the viewer.

2.2 Coherence techniques

When Ashikhmin [2] introduced the concept of coherence, he observed that the results of synthesis algorithms often contain large contiguous regions of the input texture/image when using non-parametric patch sampling. Consequently, the independent search for every patch in the input texture/image can be accelerated by using information from previously computed searches. Thus it limits the search space of a given patch to the locations of the most similar patches of its neighbors. We based our coherence-based matches refinement on the same coherence observation and developed an novel approach that is efficient with respect to both computation time and memory consumption. Tong et al. [18] also proposed a coherence technique called *k-coherence*. While this technique improved the search time, the pre-processing time and the memory consumption are major drawbacks for high-resolution video completion methods.

This paper presents an approach for the completion of video sequences that requires very low memory usage and reasonable computation time, making it usable for HD video sequences. Further, it presents a new coherence-based match refinement approach that increases the overall quality of the results by eliminating many noticeable artifacts. Unlike most of the previous works, this paper presents results on video sequences with non-stationary camera movements.

3 HIGH DEFINITION VIDEO COMPLETION

In this section, we present a new video completion approach that is able to automatically fill missing regions of HD video sequences. Section 3.1 presents the approach overview, Section 3.2 explains the dual inpainting-sampling filling-order completion approach, Section 3.3 describes the coherence-based matches refinement process, and Section 3.4 details the new localized search completion method.

3.1 Approach overview

Starting with an input video sequence V containing a missing region or hole H ($H \subset V$), our approach fills H in a visually plausible manner by copying similar patches found in the existing region E ($E = V \setminus H$), thus creating a completed video sequence V^*. This process is shown in Figure 2. In order to maintain spatio-temporal consistency, we consider the input video as a space-time volume, and thus a pixel located at (x, y) in frame t can be represented by the space-time point $p = (x, y, t)$. Consenquently, a patch w_p can be seen as a spatio-temporal cube of pixels centered at p. Table 1 summarizes the symbols used in this paper.

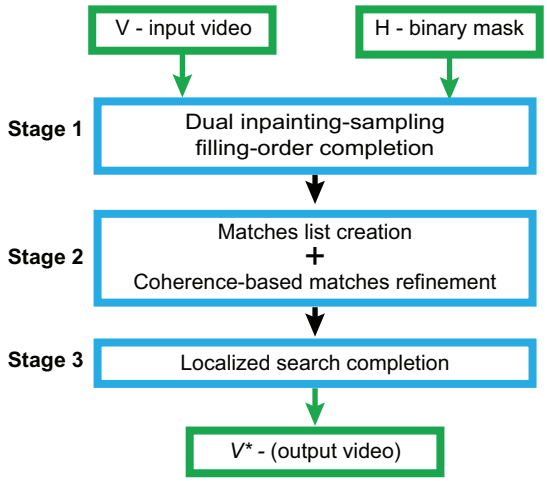


Figure 2: Schematic overview of the proposed approach

V	original video sequence
H	missing region or hole of V, $H \subset V$
E	existing region of V, $E = V \setminus H$
H^*	completed region
V^*	completed video sequence
p_l	point located at (x, y, t) at coarser resolution
w_p^l	patch centered at p_l at coarser resolution
$w_{p'}^l$	patch centered at p'_l, most similar to patch w_p^l
p_h	point located at (x, y, t) at finer resolution
w_p^h	patch centered at p_h at finer resolution
$w_{p'}^h$	patch centered at p'_h, most similar to patch w_p^h
ML	matches list
c	RGB color
S	search region centered at p'_h

Table 1: Symbols definitions

The missing region H is indicated to the system by a binary video sequence in which identified pixels are in H. The binary video sequence can be constructed using object tracking in the video sequence. Many digital motion graphics and compositing softwares already provide accurate and rapid tools to create such binary video sequences. In our experimentations, we relied on such tools to define H.

Figure 2 shows a schematic overview of our approach while Figure 3 presents the detailed steps. First, the input video is downsampled and completed by a dual inpainting-sampling filling-order completion based on the works of Wexler et al. [19, 20] using global optimization and non-parametric patch sampling (see Section 3.2). Completing the video at low-resolution with the proposed approach is efficient and provides good results. When the dual inpainting-sampling filling-order completion is finished, each patch w_p^l (centered at $p_l \in H$) is associated with its best matching patch $w_{p'}^l$ (centered at $p'_l \in E$) by creating a matches list containing

space-time pairs p_l-p'_l for every $p_l \in H$. To complete the search in a reasonable amount of time, the best matching patches $w^l_{p'}$ must be selected using approximate search and data compression methods. Thus, the $w^l_{p'}$ found might not be the best match. The second stage of the proposed method consists of an iterative coherence-based matches refinement process that improves the search results for the worst matching patches $w^l_{p'}$ (see Section 3.3). This stage is efficient, and provides significant quality improvement. Finally, the matches list is used by the localized search completion method to narrow the search space, thus enabling the completion of HD video sequences (see Section 3.4). This final stage of the method is also efficient, and it provides good results at HD resolution.

3.2 Dual inpainting-sampling filling-order completion

A visually plausible completion of a video sequence replaces the missing region H by a completed region H^* where pixels of H^* fit well within the whole video V^*. To achieve this, a video completion approach must satisfy two criteria: first, every local space-time patch of the completed region H^* must be similar to an existing patch of E, and secondly, all patches that fill H^* must be coherent with each other. Consequently, we seek a completed video V^* that minimizes the objective function stated in Equation 1:

$$Coherence(H^*|E) = \prod_{p_l \in H} \min_{p'_l \in E} D(w^l_p, w^l_{p'}), \quad (1)$$

where $D(w^l_p,\ w^l_{p'})$ is a similarity metric between two patches. The similarity value of two patches is evaluated with the Sum of Squared Differences (SSD) of color information (in the RGB color space) for every pair of space-time points contained in these patches. Wexler et al. [20] added the spatial and temporal derivatives to the RGB components to obtain a five-dimensional representation for each space-time point. In experimentations, RGB alone produced good results for most videos. Problems occured when trying to reconstruct a hidden moving object. While the technique of Wexler et al. [20] can solve these problems, it is however limited to objects with cyclic motion (i.e. like a walking person). Moreover, it requires more memory and computation time. For these reasons, we limited our problem domain to videos without occluded moving objects and chosen to use only RGB components.

The first step of the dual inpainting-sampling filling-order completion approach is to downsample V to a coarse resolution (see Figure 3, Stage 1.1). Then, before starting the completion, the values of each space-time point of H need to be initialized. Unlike Wexler et al. [20] who used random values, the proposed approach fills H using an image inpainting technique [3]

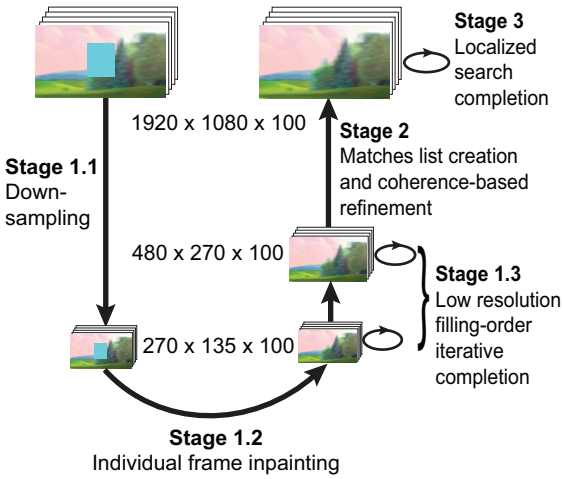


Figure 3: Steps of the proposed video completion approach

(see Figure 3, Stage 1.2). Our aim is to speed up the convergence by using the existing information around H. This initialization is done only once, prior to the first iteration of the low resolution filling-order iterative completion approach.

After the initialization, the approach performs an iterative process, improving the overall coherence of H (see Figure 3, Stage 1.3). During each iteration, the approach seeks a replacement color value for every space-time point in H in order to minimize Equation 1. Unlike previous methods, which used scan-line ordering, our approach fills H using a 3D hole-filling approach, thus ensuring that each patch w^l_p contains information that is more reliable (space-time points in E or space-time points already processed during the current iteration). Consequently, it speeds-up the convergence and reduces discontinuities near the boundaries of H. The patches can have different sizes in the spatial and temporal dimensions. Generally, we used $5 \times 5 \times 5$ patches or $7 \times 7 \times 5$ patches and we based our choice on the element structure size that needs to be completed within the video sequence.

To seek a replacement color c for a space-time point p, the approach uses a single best-matching patch $w^l_{p'}$ that minimizes $D(w^l_p,\ w^l_{p'})$. When $w^l_{p'}$ is found, the color c' is copied from space-time point p'_l to p_l. Compared to other methods that blend together several matches, using the single best-matching patch does not result in blurring artifacts and preserves film grain and noise from the original video. For these reasons, our approach uses the single best-matching patch.

To enforce spatio-temporal consistency, this iterative process is done on multiple scales using spatial pyramids (see Figure 3, Stage 1.3). Each pyramid level contains $^1/_2 \times {}^1/_2$ of the spatial resolution while maintaining the temporal resolution. The iterative process starts with the coarsest pyramid level and propagates its re-

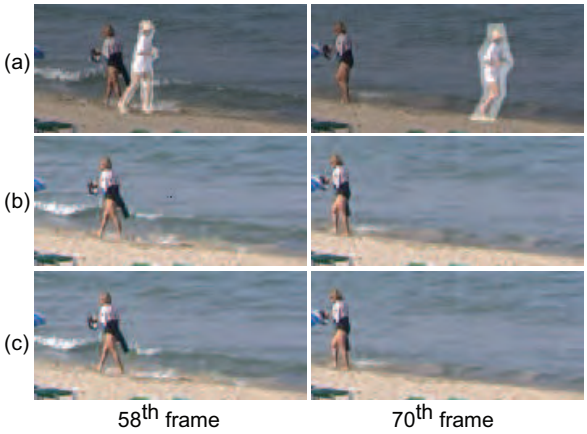


Figure 4: Comparison of the results obtained with the low-resolution video completion approach: (a) Original frames; (b) results from Wexler et al. [20]; (c) results from the proposed method

sults to finer levels. Because it involves long computation times and a lot of memory for the search structure, this iterative process is impractical at finer pyramid levels for HD videos. Therefore, the proposed approach stops the iterative process when it reaches a fixed resolution (typically 480×270).

The proposed dual inpainting-sampling filling-order completion approach produces results with a quality equivalent to the results of Wexler et al. [20], but within much less time. Figure 4 shows the completion results of the "Jogging lady" sequence of Wexler et al. [20] and ours. Wexler's approach took one hour per iteration at the finest resolution level while our approach took less than four minutes per iteration.

3.3 Coherence-based matches refinement

When Stage 1 is over, each patch $w_p^l \in H^*$ has a corresponding patch $w_{p'}^l$. Each space-time point p_l is associated with its corresponding p'_l and the pairs are stored in a matches list *ML*. During the high-resolution completion iterative process, *ML* enables the approach to narrow the search space to only sub-regions of *E*. As a reminder, our key observation is that, for a patch w_p^l at coarser resolution with its most similar patch $w_{p'}^l$, the most similar patch of the corresponding patch w_p^h at finer resolution is likely to be found near p'_h (see Figure 1).

For efficiency reasons, optimization methods such as principal component analysis (PCA) and approximate nearest neighbors search (ANN) [1] are used in Stage 1. While these methods are essential to achieving acceptable search times, they often provide matches $w_{p'}^l$ that do not minimize Equation 1. Consequently, *ML* needs to be refined during Stage 2 (see Figure 3, Stage 2) to have better matching patches $w_{p'}^l$.

(a) Original frame (b) Completed HD frame without *ML* refinement (c) Completed HD frame with *ML* refinement

Figure 5: Impact of the *ML* refinement iterative process on high-resolution video completion results: (a) Original frame; (b) completed frame without *ML* refinement; (c) completed frame with *ML* refinement. The frames were cropped to better show the missing and completed regions

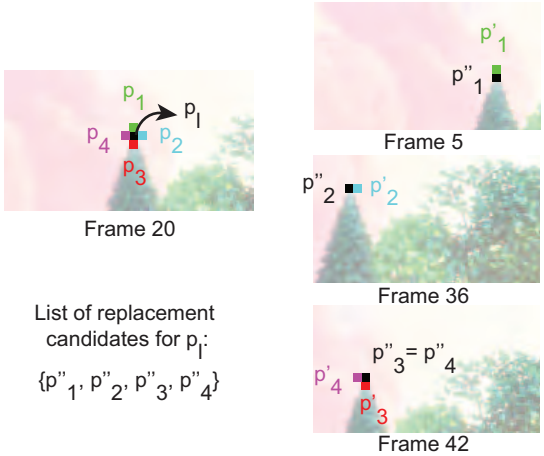


Figure 6: Coherence-based matches refinement process

The information contained in *ML* must be reliable in order for visually plausible results to be possible with the localized search completion iterative process. The patches w_p^l and $w_{p'}^l$ must be highly similar for every pair p_l-p'_l of *ML*, otherwise, the approach will narrow the search space to a region where it is less likely to find the best matching $w_{p'}^h$. Figure 5 (a, b) shows an example where the information contained in *ML* is not reliable. As can be seen, there are many visible artefacts such as the centered window and the left building edge.

To find better matching patches $w_{p'}^l$, we take advantage of the concept of coherence. First, the approach calculates the distance (L_2 norm of uncompressed data) of patches w_p^l and $w_{p'}^l$ for each pair p_l-p'_l from *ML*. Then an iterative process refines pairs with distances higher than a given threshold. During the first iteration, this threshold is set such that 15% of the pairs are refined. After each iteration, this threshold is reduced by 20% of its initial value. For each pair p_l-p'_l above the threshold, the approach seeks for a replacement p'_l that decreases $D(w_p^l, w_{p'}^l)$. Instead of using a brute force approach that searches the entire video sequence, the search is restricted around the best matching patches of

p_l neighbors. An example is shown in Figure 6. The four neighbors of p_l are considered: top, right, bottom, and left; respectively p_1, p_2, p_3, and p_4. For the top neighbor (p_1) the approach considers its previously calculated best matching point p'_1, then from p'_1, its bottom neighbor p''_1 is considered. The L_2 norm is computed between patches p_l and p''_1, and if the norm is lower than the current value, p'_l is replaced by p''_1 and the color from p''_1 is copied to p_l. This process is repeated for p_2, p_3, and p_4. If there is no good replacement, the pair p_l-p'_l is left unchanged, and is considered in the next iteration.

When considering the top neighbor p_1, instead of searching anywhere around its best matching point p'_1, only its bottom neighbor (p''_1) is considered. The rationale behind this is that several successful approaches use large primitives such as fragments or epitomes. When considering larger primitives, the bottom neighbor (p''_1) is the one that would be copied on top of p_l. This effectively reduces the search to only four points (p''_1 to p''_4). To even further reduce the number of points to test, each neighbor p_1 to p_4 is considered only if the L_2 norm of a pair, for example, p_1-p'_1, is below the current threshold. This is a very rapid test since the value is already computed and stored in the *ML*. Since there is a maximum of only four potential points to consider as opposed to the millions from the whole video sequence, this process is extremely fast. Figure 7 shows an example of the *ML* coherence-based matches refinement process that minimizes the distance of w_p and $w_{p'}$ for each pair p_l-p'_l in *ML*. The *ML* refinement provides a significant quality improvement (as shown in Figure 5) within a few seconds.

3.4 Localized search completion

This section presents the proposed approach for completing missing regions of video sequences at HD resolutions. As stated earlier, current exemplar-based methods are unpratical to complete HD video sequences because best match searches require excessive amount of memory and computation time. Many attempts have been made to accelerate this search with optimization methods such as ANN and dimensionality reduction methods such as PCA, but the structures needed for these optimization methods require too much memory for HD video sequences. Instead of accelerating the best match search, the proposed method narrows the search space at HD resolution using information from coarser resolutions.

Before the localized search completion process starts, the information contained in *ML* must to be scaled up to the finer resolution (see Figure 8). For each space-time point $p_h \in H$ at a finer resolution, its corresponding low-resolution p_l is found as well as the space-time location p'_l associated with it. The space-time location

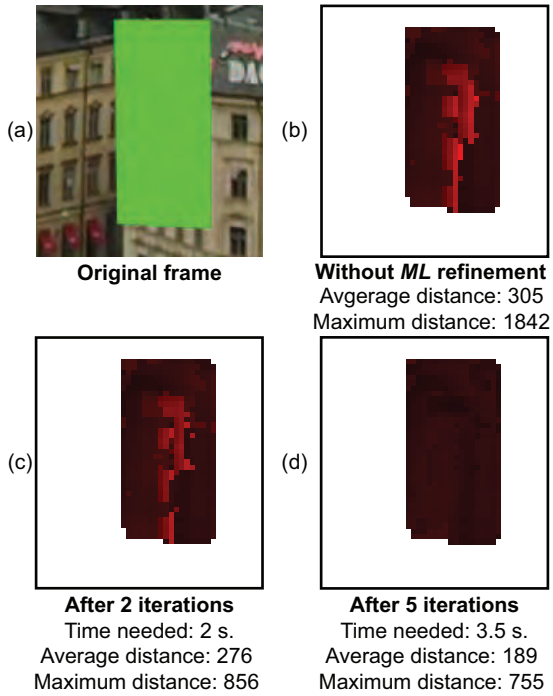


Figure 7: Impact of the *ML* coherence-based matches refinement process: (a) original frame; (b) distance of w_p and $w_{p'}$ for each pair p_l-p'_l after *ML* creation; (c) distances after two iterations of the *ML* refinement process; (d) distances after four iterations of the *ML* refinement process. The frames were cropped to better show the missing and completed regions

p'_l is then scaled up to a finer resolution resulting in p'_h. The pair p_h-p'_h is then added in a new matches list *MLH* which will be used by the localized search completion process to narrow the search space.

The main steps of the localized search completion process are similar to those of the low-resolution process: using a 3D hole-filling approach, the method seeks a

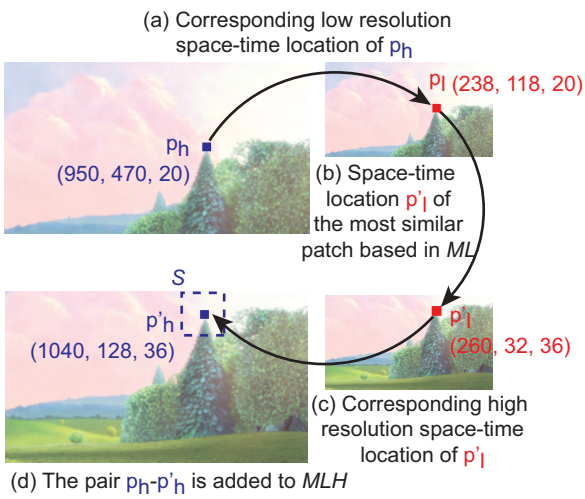


Figure 8: Creation of *MLH* based on *ML*

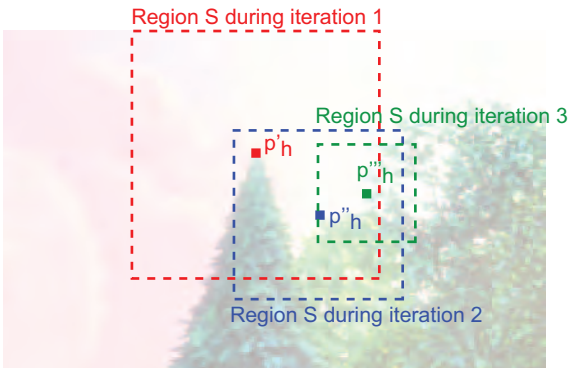


Figure 9: Locations and sizes of search regions S for the first three iterations of the localized search completion process

replacement color c' for every $p_h \in H$ using a single best-matching patch $w^h_{p'}$. However, instead of searching through the entire video sequence using a brute force algorithm or expensive search structures, the approach only searches in a small sub-region S, based on the information from *MLH*. For each space-time point $p_h \in H$, the approach first looks in *MLH* and seeks for its associated p'_h. Then, a small region S centered at p'_h is selected. Next, the approach searches only in S for the best-matching patch $w^h_{p''}$ (located at $p''_h \in S$) and the color c_h is replaced by c''_h. If p'_h and p''_h are different, the pair p_h-p'_h from *MLH* is replaced with the pair p_h-p''_h. In the next iteration of the localized search completion process, the sub-region S will be recentered around this updated space-time location. During the first iteration of the localized search completion process, the window size of sub-region S is 17×17 pixels. This window size is then decreased after each iteration (13×13, 9×9, 5×5). Figure 9 shows an example of the location changes and size decreases of a search region S for three iterations.

Obviously, the search time is dramatically reduced when using *MLH* to narrow the search space, as compared to using methods such as ANN and PCA. When using the proposed *MLH* technique, less than a thousand patches are searched for each p_h compared to the tens of millions of patches from the whole video. Moreover, the computation time for the creation and the refinement of *ML* and *MLH* is shorter than the time needed for the creation of the structures used by ANN and PCA. Another important advantage of the proposed method is that *MLH* requires much less memory than typical search structures, such as ANN. Finally, the proposed *MLH* search does not rely on compressed data, and thus can provide better matches.

4 RESULTS AND DISCUSSION

Figure 10 shows the completion of the "Station" sequence and Figure 11 shows the completion of the "Race to Mars" sequence. The main challenge of these sequences is the constant motion of the camera. The "Station" sequence contains a constant zooming motion while the "Race to Mars" sequence contains complex rotating and panning motions. Video sequences with such motions cannot be handled by video completion techniques using a static background mosaic because the size and orientation of the objects contained in the background are not constant during the entire video sequence.

It can be seen in Figure 10 that the proposed method works well with large missing regions. Figures 10 and 11 demonstrate that the proposed methods produce good results for missing regions containing stochastic texture as well as salient structure. Since state of the art papers introduced in Section 2 show results with resolutions ranging from 320×240 to 640×480, it is not possible to compare the quality of our results with other techniques. Therefore, we used a structural similarity method (SSIM) [13], a full reference metric, to measure the quality of our results at high-resolution. Even though SSIM is generally used to evaluate video compression methods, it can also be used to measure the similarity between a reference sequence and a completed sequence. Figure 12 shows the completion of the "Old town cross" sequence. Considering only the pixels in the missing region instead of all the pixels from the full frames of the sequence, the average SSIM index is 90.63. Since the completed region does not need to be exactly like the reference region, as long as the region is completed in a visually plausible manner, this SSIM index is good.

Table 2 shows a comparison of the proposed approach with earlier works based on different criteria (some were taken from Shih et al. [15]):

- **Missing region specification**: how the user interacts with the method to specify the missing region;
- **Exemplar-based approach**: what type of completion method is used;
- **Camera motion**: video sequences with stationary or nonstationary camera;
- **Maximum resolution**: the highest resolution of the video sequences presented in the paper.

All completion methods use an exemplar-based technique with different variations. Most of the completion methods only use video sequences taken with a stationary camera to test their algorithm. Patwardhan et al. [12] present results with a non-static camera, but the camera motion is always parallel to the projection plane. Thus, Patwardhan et al. [12] do not deal with changes in size, perspective, nor zooming. Only Shih et al. [15] and the proposed method present results with

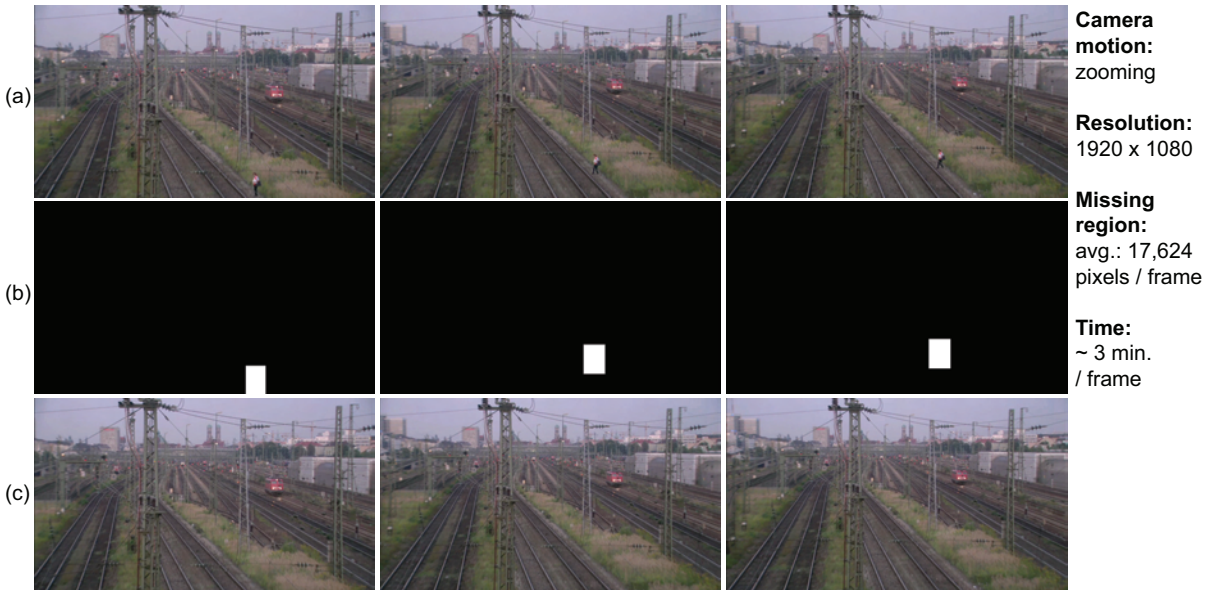


Figure 10: Results for the "Station" sequence : (a) Original frames; (b) missing regions; and (c) completed frames. Frames from `https://cs-nsl-wiki.cs.surrey.sfu.ca/wiki/Video_Library_and_Tools`

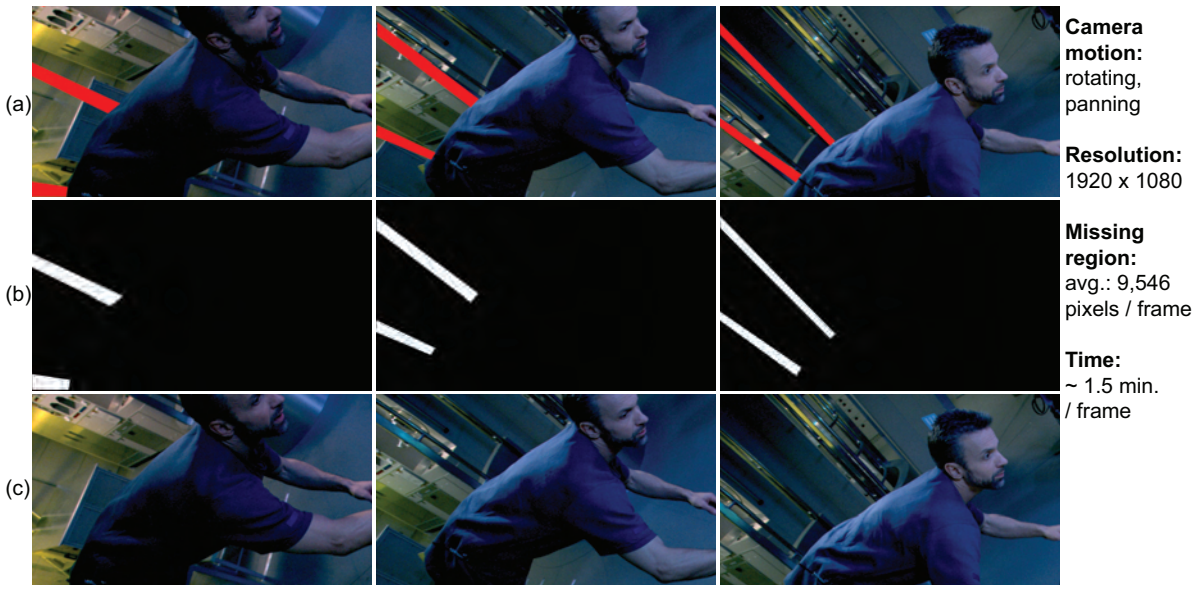


Figure 11: Results for the "Race to Mars" sequence (frames were cropped to better see the regions): (a) Original frames (with unwanted wires highlighted in red); (b) missing regions; and (c) completed frames. Frames from "Race to Mars", a courtesy of Galafilm and Discovery Channel Canada

different camera motions such as zooming, rotating, and panning. Finally, the main advantage of the proposed method over previous works is the maximum resolution it can handle. The proposed method handles HD video sequences while the highest resolution of all previous works from Section 2 is only 640×480, which is more than a six-fold improvement over state-of-the-art exemplar-based methods.

5 CONCLUSION

We have presented a video completion method that can handle much higher resolutions than previous work. The proposed method is based on three new approaches: a dual inpainting-sampling filling-order completion, a new coherence-based matches refinement, and a new localized search completion approach. Together, these three approaches solve the memory consumption and computation time problems for the completion of HD video sequences. Furthermore, the quality of the results generated by our method compares fa-

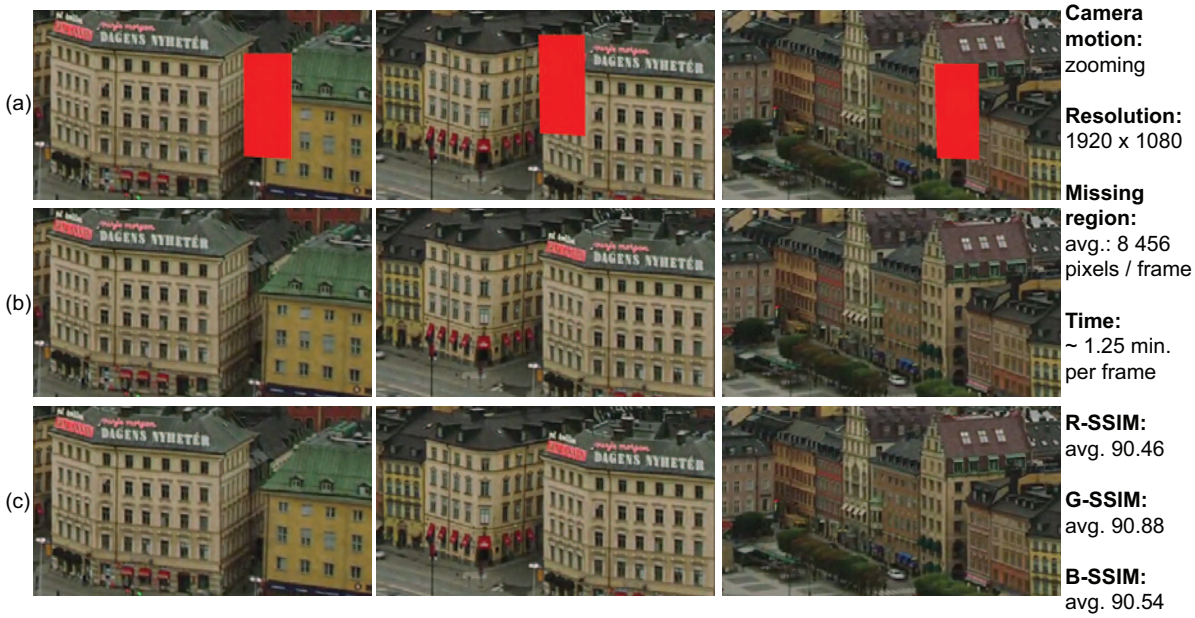


Figure 12: Results for the "Old town cross" sequence (frames were cropped to better see the regions): (a) Frames with a synthetic object; (b) completed frames; and (c) clean frames. Frames from `https://cs-nsl-wiki.cs.surrey.sfu.ca/wiki/Video_Library_and_Tools`

	Criteria			
Related works	**Missing region(s) specification**	**Exemplar-based approach**	**Camera motions**	**Maximum resolution**
Kamel et al. [7]	User provided mask	Standard	Static	80×110
Shih et al. [15]	Bounding box given by user, missing region is tracked	Improved patch-matching	Static, non-static	320×240
Liu et al. [10]	User provided mask	Motion fields and colors	Static	320×240
Xiao et al. [21]	User provided mask	Motion similarity and colors	Static	384×192
Shiratori et al. [17]	User provided mask	Motion fields	Static	352×240
Wexler et al. [20]	User provided mask	Motion similarity and colors	Static	360×288
Koochari and Soryani [8]	User provided mask	Standard	Static	540×432
Patwardhan et al. [12]	User provided mask	Motion inpainting and priority based texture synthesis	Static, non-static	640×480
Herling and Broll [5]	Rough selection by user, missing region is tracked	Combined pixel-based approach	Static, non-static	640×480
The proposed approach	User provided mask	dual inpainting-sampling filling-order completion, coherent and localized search	Static, non-static	1920×1080

Table 2: Comparison of the proposed method with previous works

vorably to previous works and allows for a significant increase of the resolutions that can be completed.

The proposed coherence-based match refinement is promising as it could be applied at various steps of several video completion approachs. Future work will involve an investigation of when the coherence approach provides the best improvements: between each iterations; between each resolution levels; at coarser or finer resolutions; etc. As they are used in the proposed method, the coherence-based matches refinement and localized search completion consider a fairly limited number of patches. Therefore, the search could stop in a local minimum while there are better matches elsewhere in the video. Future work should look at appropriate techniques to expand the search to other locations that are likely to contain good matches.

6 ACKNOWLEDGMENTS

We would like to thank the Fonds québécois de la recherche sur la nature et les technologies (FQRNT),

Mokko Studio inc., and the École de technologie supérieure for funding this project. We would also like to thank all members of the Multimedia Lab for their reviews. Finally, we would also want to thank Galafilm and Discovery Channel Canada for the "Race to Mars" video sequence.

REFERENCES

[1] Arya, S., Mount, D.M.: A library for approximate nearest neighbor searching (2010). URL `http://www.cs.umd.edu/~mount/ANN/`

[2] Ashikhmin, M.: Synthesizing natural textures. In: 2001 ACM symposium on Interactive 3D graphics, pp. 217–226. ACM (2001)

[3] Bertalmio, M., Bertozzi, A., Sapiro, G.: Navier-stokes, fluid dynamics, and image and video inpainting. In: Proc. Conf. Comp. Vision Pattern Rec., pp. 355–362 (2001)

[4] Cheung, V., Frey, B.J., Jojic, N.: Video epitomes. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, vol. 1, pp. 42–49 (2005)

[5] Herling, J., Broll, W.: High-quality real-time video inpaintingwith pixmix. IEEE Transactions on Visualization and Computer Graphics **20**(6), 866–879 (2014)

[6] Jia, J., Tai, Y.W., Wu, T.P., Tang, C.K.: Video repairing under variable illumination using cyclic motions. IEEE Transactions on Pattern Analysis and Machine Intelligence **28**(5), 832 –839 (2006)

[7] Kamel, S., Ebrahimnezhad, H., Ebrahimi, A.: Moving object removal in video sequence and background restoration using kalman filter. In: Int. Symp. on Telecommunications 08, pp. 580–585 (2008)

[8] Koochari, A., Soryani, M.: Exemplar-based video inpainting with large patches. Journal of Zhejiang University - Science C **11**(4), 270–277 (2010)

[9] Ling, C., Lin, C., Su, C., Liao, H.Y., Chen, Y.: Video object inpainting using posture mapping. In: Proc. IEEE Int. Conf. on Image Processing, pp. 2785–2788 (2009)

[10] Liu, M., Chen, S., Liu, J., Tang, X.: Video completion via motion guided spatial-temporal global optimization. In: Proc. ACM Multimedia, pp. 537–540 (2009)

[11] Patwardhan, K., Sapiro, G., Bertalmio, M.: Video inpainting of occluding and occluded objects. In: Proc. IEEE Int. Conf. on Image Processing, vol. 2, pp. 69–72 (2005)

[12] Patwardhan, K.A., Sapiro, G., Bertalmio, M.: Video inpainting under constrained camera motion. IEEE Transactions on Image Processing **16**(2), 545 –553 (2007)

[13] Sheikh, H., Sabir, M., Bovik, A.: A statistical evaluation of recent full reference image quality assessment algorithms. IEEE Transactions on Image Processing, **15**(11), 3440 –3451 (2006)

[14] Shen, Y., Lu, F., Cao, X., Foroosh, H.: Video completion for perspective camera under constrained motion. In: Proc. of the 18th Int. Conf. on Pattern Recognition, vol. 3, pp. 63–66 (2006)

[15] Shih, T., Tang, N., Hwang, J.N.: Exemplar-based video inpainting without ghost shadow artifacts by maintaining temporal continuity. IEEE Transactions on Circuits and Systems for Video Technology **19**(3), 347–360 (2009)

[16] Shih, T.K., Tang, N.C., Yeh, W.S., Chen, T.J.: Video inpainting and implant via diversified temporal continuations. In: Proc. of the 14th annual ACM int. conf. on Multimedia, MULTIMEDIA '06, pp. 965–966. ACM (2006)

[17] Shiratori, T., Matsushita, Y., Tang, X., Kang, S.B.: Video completion by motion field transfer. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, vol. 1, pp. 411–418 (2006)

[18] Tong, X., Zhang, J., Liu, L., Wang, X., Guo, B., Shum, H.Y.: Synthesis of bidirectional texture functions on arbitrary surfaces. ACM Trans. Graph. **21**, 665–672 (2002)

[19] Wexler, Y., Shechtman, E., Irani, M.: Space-time video completion. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, vol. 1, pp. 120–127 (2004)

[20] Wexler, Y., Shechtman, E., Irani, M.: Space-time completion of video. IEEE Trans. on Pattern Analysis and Machine Intelligence, **29**(3), 463–476 (2007)

[21] Xiao, C., Liu, S., Fu, H., Lin, C., Song, C., Huang, Z., He, F., Peng, Q.: Video completion and synthesis. Computer Anim. Virt. Worlds **19**, 341–353 (2008)

[22] Zhang, Y., Xiao, J., Shah, M.: Motion layer based object removal in videos. In: 17th IEEE Workshops on Application of Computer Vision, vol. 1, pp. 516 –521 (2005)

Evaluating Stereoscopic Visualization for Predictive Rendering.

Fernando da Graça
Center for Robotics
MINES ParisTech,
PSL-Research University
60 Boulevard Saint Michel
75006, Paris, France
fernando.graca@mines-paristech.fr

Alexis Paljic
Centre for Robotics
MINES ParisTech,
PSL-Research University
60 Boulevard Saint Michel
75006, Paris, France
alexis.paljic@mines-paristech.fr

Emmanuelle Diaz
R&D Department,
Cognitive Science and Human Factors
PSA Peugeot Citroën
2, route de Gisy
78943,
Vélizy-Villacoublay Cedex, France
emmanuelle.diaz@mpsa.com

ABSTRACT

The context of this work is predictive rendering; our objective is to previsualize materials based on physical models within computer graphics simulations. In this work we focus on paints constituted of metallic flakes within a dielectric binder. We want to validate a "virtual material workshop" approach, where a user could change the composition and the microstructure of a virtual material, visualize its predicted appearance, and be able to compare it to an actual sample. To do so, our methodology is to start from Scanning Electron Microscopy (SEM) imaging measures on an actual sample that allowed us to characterize two metrics: flake size and flake density. A statistical model based on those measures was then integrated in our spectral rendering engine using raytracing and photon mapping, with an off axis-frustum method to generate stereoscopic images for binocular visualization. Our objective is twofold: 1) perceptually validate our physical model, we evaluate if the virtual metric perceptually corresponds to the real metric of the real samples; 2) evaluate the contribution of virtual reality techniques in the visualization of materials. To do so, we designed a user study comparing photographs of car paint samples and their virtual counterpart based on a design of experiments. The observers evaluated the visual correspondence of different virtual materials generated from microstructures with varying metric values. The results show a perceptual correspondence between real and virtual metrics. This result has a strong impact: it means that for a desired appearance the proposed models correctly predict the microstructure. The second result is that stereoscopy improves the metric correspondence, and the overall appearance score.

Keywords

virtual-reality, predictive rendering, visual perception, complex materials, metallic paints, microstructure, statistical model

1 INTRODUCTION

The study of visualization quality is a crucial step to accurately represent digital prototypes. Manufacturing departments understand this problematic, since they rely on digital representations of the end product to make critical choices about its final appearance. Furthermore, it gives them the freedom to comprehensively explore (simulate) a large variety of materials without the need to manufacture the actual object. However, a correct representation of real materials still remains a challenging task. To this end, researchers focused on developing Predictive Rendering (PR) techniques to reduce de gap between the observations of a physical object and its virtual replica.

According to Wilkie et al. [1] PR, as opposed to believable rendering, is a field of research that aims at creating physically correct computer images [2, 3]. The objective is to predict the true visual appearance from a virtual reflectance model, which takes into consideration the physical parameters of the actual material. If such a tool was to be mastered, it would allow to design a virtual material and simulate its visual appearance iteratively. Then, when the desired appearance is obtained for the digital prototype, the actual equivalent object could be produced with the same set of param-

eters that were used as input for the virtual reflectance model. This process is meant to lead to better design decisions at a lower cost. Predictive rendering has a very strong potential in various application domains, ranging from manufacturing industries (automotive, aeronautics), cosmetics to architecture (material design).

On the other hand, complex materials such as automotive paints are very challenging to simulate, due to their spatially varying reflectance. To this end, we argue that VR tools are indispensable to fully explore the visual aspect of such materials. Virtual Reality is used in the industry for product design or industrial process validations. Through the use of stereoscopic displays, user motion capture, motion parallax, VR allows for a user to feel immersed in a 1:1 scale virtual environment, and to observe objects and interact with them. In particular, binocular vision and motion parallax are necessary to provide the human visual system with valuable depth and shape cues. These are key characteristics of VR that are necessary to ensure "human in the loop" simulations. In order to combine the physical and visual validity of Predictive Rendering and perception cues provided by Virtual Reality, the need for stereoscopy and motion parallax for predictive rendering simulations is being expressed by the industry.

Predictive rendering approaches require perceptual validations that take into account the human visual system in order to be valid. The field of research for such validations is large and we are only starting to draw the boundaries within which virtual material samples are representative of real material samples. Indeed, image quality perception can depend on a lot of parameters that appear at several stages of the process. One should, at least, consider the following: the human user (visual acuity, individual color perception, visual fatigue), the technical setup (display calibration, display resolution, luminance), the sensory motor inputs and outputs (use of stereoscopy, motion parallax, user's ability to manipulate the virtual material sample), and the rendering engine itself (light-matter interaction models, material models). In this context, this work is part of an iterative validation process in a research project of a predictive rendering engine in which the objective is to link microstructure and appearance. In this work, our objective is twofold: 1) evaluate the pertinence of a "virtual material workshop" approach where a user could change the composition and the microstructure of a virtual material, visualize its predicted appearance, and be able to compare it to an actual sample. To do so, we propose to use a microstructure model [4] based on measures of actual material samples presented in section 2.2; 2) evaluate the role of stereoscopy on perception of materials that depict binocular differences such as automotive paints with metallic flakes. For this purpose, our methodology evaluates, through a user study, the visual agreement between the observation of the computer generated object and the actual object. We introduce a novel approach in Computer Graphics (CG) domain to render virtual materials by using the microstructure formulation of the real material, see figure 2.

We begin the paper with a survey of related work in section 1.1. In section 2 we describe the simulation of virtual automotive paints using a microstructure model. Then, in section 3 we describe the experimental setup of the virtual scene, and section 4 the design of experiment to evaluate the response of the observers. In the section 5 we present the results of the measured data. The experimental results are analyzed in the section 6. Finally, in section 7 we present our conclusions, and we address some aspects for future work to complement our findings.

1.1 Related Work

In this section, we first propose an overview of the existing CG (Computer Graphics) methods for computer generated images of materials with nano/macro inclusions such as car paints with flakes. We then explore the existing literature on the role of stereoscopy on the perception of surface aspect.

In the CG domain, several researches have proposed different methods to simulate car paint models. These models are based on Bidirectional Reflectance Distribution Function [5], which represents how the surface reflects the incident light at different angles. The distribution of the reflectance of the light can be captured by optical measurement devices [6]. Then, the obtained data is used to derive reflectance models that represent the appearance of the physical material [7]. In addition, we can also consider the Bidirectional surface scattering distribution function (BSSRDF) models [8] that takes into account the scattering of the incident beam of light in the interior of the material.

Generally speaking, we can distinguish two groups: analytical, and data-driven models. In the first, the user tweaks several parameters until he achieves a visual aspect that is similar to the real paint. Durikovic et al. [9] model the geometry of the flakes inside the paint film. Their system is capable of generating stereoscopic images, and it allows to define the parameters for the random distribution of the position of flakes and their orientation. The approach of Ershov et al. [10] is based on reverse engineering, the appearance attributes such gloss are added to the physical model by adjusting parameters. The inconvenient of these models is the amount of parameters that are necessary to represent the car paint appearance.

In the second approach, Günther et al. [11] developed an image based acquisition setup to measure the Bidirectional Reflectance Distribution Function (BRDF) of

a car paint. To this BRDF they add the sparkle simulation. The distribution of the the flakes is stored as a texture. Rump et al. [12] used a similar approach as the previous one. The difference is that, instead of using a sparkle simulation with the BRDF, they capture the spatially varying appearance (sparkle effect) using Bidirectional Texture Function (BTF) measurements. In addition, they store and simulate directional visual effects. Sung et al. [13] determine individual flake orientations in the car paint by using confocal laser scanning microscope. They capture the angular dependent reflectance using a goniometer. Then, they use these measurements to build a reflectance model. In the context of this work, we need to evaluate the perception limits in a VR environment, when the observer perceives the aspect of the reflectance models using binocular vision. A human observer perceives the binocular summation of the left and right image. In the combination of the two images, we can identify two cases binocular fusion and rivalry. In the first case, if two retinal points are sufficiently similar, a binocular combination occurs. In contrast, when two retinal points are very distinct, the observer perceives a fluctuation between the left and right images, that is, a failed fusion occurs. This phenomenon is known as binocular rivalry [14].

Most of the work was done on the role of stereoscopy for the perception of gloss. In the literature the gloss is defined to be a global property of the surface aspect [15, 16]. Glossy surfaces reflect the incident light to a particular outgoing direction, characterized by a specific angular interval. The perception of specular reflections is an important cue to evaluate the glossiness of the materials. In addition, there is an influence of binocular cues such as highlight disparities on the perception of gloss [17]. The experimental results of Wendt et al. [18] show an improvement of gloss perception when highlights cues disparities are taken into account. The experimental results of the work of G. Obein et al. [19] suggest also that the binocular vision helps for judgment of high gloss samples. They found that with binocular vision the sensitivity to gloss is higher than the monocular vision, for high gloss levels. Sakano et al. [20] examined the effects of the combination of self-motion and contingent retinal-image motion (motion parallax) on perceived glossiness. When the observer was able to move his head, a stronger glossiness was perceived than when both the observer and the stimulus were static. From their experimental results, they found that the glossiness under the monoscopic condition was underestimated compared to stereoscopic condition. The glossiness under static (head not moving) condition was underestimated compared to dynamic condition. Knill et al. [21] study the combination of different cues for slant perception. At low slants, observers use more the binocular cues than the texture. At slants of 50 and 70, the subjects do better slant judgments using the texture information of the image. As the slant increases the observers give more attention to texture information.

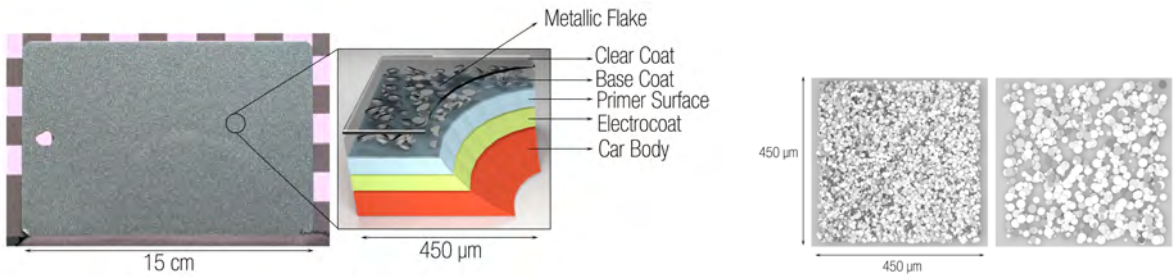
2 MATERIAL SIMULATION

2.1 Physical Plate

This study focus on grey automotive paints with metallic flakes up to 30μm. Figure1(a) shows a cross section of the studied actual sample of a car paint. Typically an automotive paint is made of four layers: clear coat, base coat, primer surface, and electrocoat. The metallic flakes are made of aluminum, and are distributed with different orientations on the primer surface of the plate at different depths, and the clear coat is transparent. The amount, distribution, and the orientation of the flakes are controlled by a milling machine. Due to their size, and orientation, these metallic pigments convey distinctive visual appearances to the object such as sparkle, and directional visual effects. At the macroscopic scale of visualization, the appearance of these nonuniform paints depend on the lighting conditions (directional/diffuse), distance and angle of observation, orientation, diameter, and density of the flakes [22, 23, 24]. The nanoscopic effects such as the chemical composition, rugosity and clustering effects of the flakes can also influence the visual appearance. In this work we only consider the macroscopic effects for which we use geometric optics models.

2.2 Stack Model

As for the models of the microstructure, we can distinguish two cases: a quasistatic, where the particles are smaller than the wavelength, consequently the human eye cannot perceive each particle. We can obtain the macroscopic visual aspect of the microstructure, which is visible by the human eye, by using homogenization methods on agglomerations of nanoscopic particles. In this work, we are interested in the second case, geometric optics, where the particle size is larger than electromagnetic visible wavelength, which is suitable to be directly used in our render engine. In this case the microstructure simulates the distribution of microscopic particles, such as metallic flakes, according to the analyses of the actual object. Morphology and statistical analysis of Scanning Electron Microscopy (SEM) images of real objects is used to create virtual microstructures that geometrically corresponds to the one of the real material. These microstructures are then used to simulate the optical behavior between the different particles. Secondly, it allows the user to modify the microstructure in order to tune the visual appearance keeping the physical feasibility, and therefore the manufacture of the virtual material, see figure 2. The stack model [4] used in our render engine distributes the metallic flakes on the surface. Though the complex



(a) (b)

Figure 1: On the left of the image (a), an example of a real car paint plate with a diagram of a typical coating paint. The flakes have typically the shape of a disk with a diameter of 5μm to 50μm, in the case of a car paint. The surface can contain rugosity and an interference film. The flakes are imbedded in the base coat with different orientations and at different depths. The figure (b) depicts two microstructures generated by the stack model. On the left a microstructure with metallic flakes of size 15μm, and on the right with metallic flakes of size 30μm.

geometry of a real metallic flake, the virtual metallic flakes were modelized with a flat cylinder shapes whose height and radius is parameterized, see figure 1(b). The statistical variation of dispersion, size, orientation is measured on the SEM images of the real plate using different morphological analyses. Then, the model simulates the clusters of flakes by using a 2D Poisson point process. Due to the large number of flakes on the surface. The stack model generates continuous microstructures of size $450\mu\text{m} \times 450\mu\text{m} \times 20.24\mu\text{m}$. The generation of plates with different flake densities and radius sizes is controlled by two parameters of the stack model. For each microstructure we have information about the radius size, the orientation, position, and amount of flakes. To generate a virtual plate with $15\text{cm} \times 10\text{cm}$, a set of 332×240 microstructures is necessary to fill the surface of the virtual plate. To avoid the large amount of geometry in our simulations, we could consider the usage of a set of textures encoding different types of information about the flakes, for instance the normal vector, and depth.

2.3 Rendering of the Virtual Model

For the optical simulation of the virtual object we used our spectral render engine raytracing with photon mapping. The scene was rendered within the visible spectrum interval $[380\text{nm} - 780\text{nm}]$ with a wavelength step of 5nm. This interval corresponds to the range of wavelengths that the human eye is sensitive. Then, we used a virtual scene with the same light conditions that were used in the real room. We associate for each plate the optical constants n and k of aluminium. In the virtual scene we used the spectrum of SOLUX $4700K$ light. The index of absorption was found in a prior experience with paired comparison. The observers were asked to evaluate which of the two virtual images was the closest to the photograph. The found absorption coefficient is $k = 6$. Furthermore, we noticed that the thickness and the absorption coefficient have an important role in the brightness of the metallic flakes. We used a virtual plate with three layers. The base layer is a black surface to minimize the back-surface reflection. The second layer, the binder, contains the microstructure of the metallic flakes, which was generated by the stack model. Finally, the clear coat layer imparts a glossy appearance to the plate. The virtual samples were defined thanks to a design of experiments. We have two factors the flake size and the flake density. A surface response design was chosen in order to evaluate the influence of the main effects but also the potential non linear effects and the interaction between factors. 13 virtual samples were built following a central composite design with replicated centered points.

3 SETUP

In this section we describe the configuration of the virtual scene. Figure 3(a) represents the virtual scene used in the experiments. We used a directional isotropic light, which is emitted from the top. The virtual cameras are placed at a distance of 77cm from the plate. The stereo cameras have an interocular distance of 6.5cm and the cyclopean camera is placed in the middle of the left and right cameras. Figure 3(b) shows the distance of observation, and the display area to visualize the virtual plates. The field of view, fov, of the camera was calculated using the observation distance, a, and the width of the area of projection, b, $fov = 2 \times \arctan\left({}^{2a}/_{b}\right)$.

4 USER STUDY

We designed a user study with 26 subjects using experimental design theory to optimize the number of trials run in order to obtain valid results using a minimum parameters variation of flake density and flake radius, see figure 5 (c). During the experiment, the subjects sat in a dark room, facing a stereoscopic screen. They observed two series of plates (stereoscopic/monoscopic),

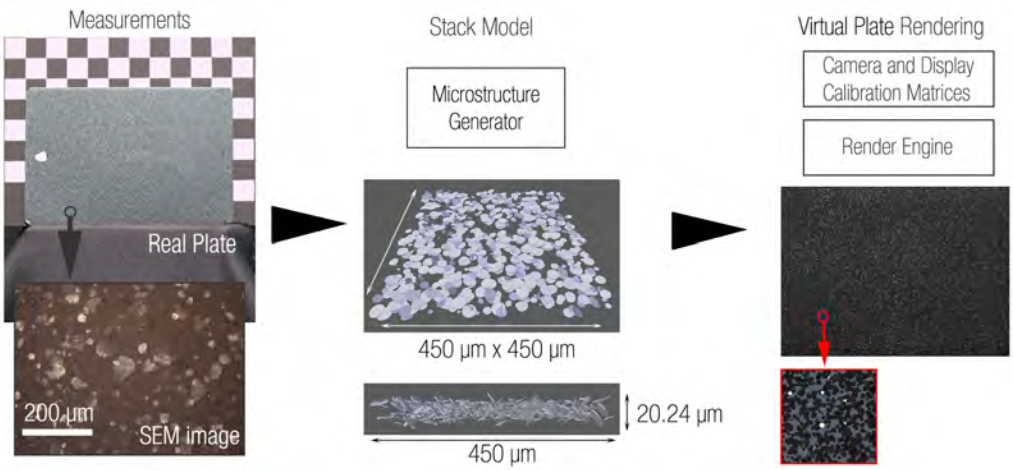
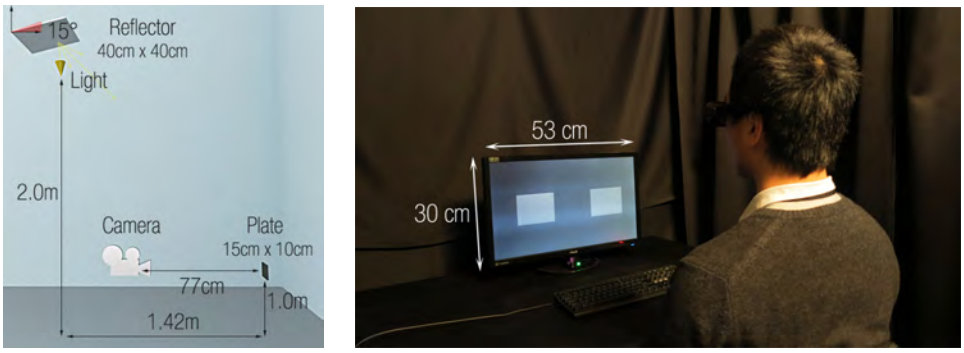


Figure 2: Through morphological and statistical analysis of SEM images, a set of microstructures are generated in order to fill the surface of the virtual plate. Then, the render engine computes the appearance of the virtual plate using the information of the microstructures, and the colorimetric calibration matrices of the photographic camera and the screen display used during the experiment.



(a) (b)

Figure 3: The image (a) is a schematic diagram used for the virtual scene. (b)The physical observation conditions to visualize the virtual plates.

Factor	Low Level	High Level	Physical Value
Flake Density	3.5	9.5	1.0
Metallic Flake Radius	15μm	45μm	30μm

Table 1: Based on the morphological and statistical analysis of the actual automotive paint, the flake radius is 30μm, and flake density is 1.0 (this value is unitless). The minimum and maximum range for flake density, and flake radius were found empirically. Within each interval, 13 values were chosen through Central Composite Design.

each one with different flake density and flake radius, see table 1. Subjects evaluated the similarity of the virtual plate to the photograph of the actual plate by using a scale from 0 (equal to the reference plate) to 10 (different from the reference plate). After an answer was given, the subject changed to the next plate with different radius and density of metallic flakes. For each trial, the presentation order of the stimuli follows a Williams Latin Square Design. The main experiment was preceded by a practice trial of two stimuli to gain familiarity with the experience.

The subjects used stereoscopic active shutter glasses during both monoscopic and stereoscopic conditions. For each trial, the subjects observed two sequences of 13 stereoscopic images. The first sequence corresponds to the monoscopic-condition, while the second corresponds to the stereoscopic case. In all experiments, the subjects used stereoscopic active shutter glasses NVIDIA 3D Vision 2 during both monoscopic and stereoscopic conditions. For the monoscopic conditions the same image was displayed for both eyes. The images were displayed using an active stereo ASUS VG248QE display. The stimulus was displayed with

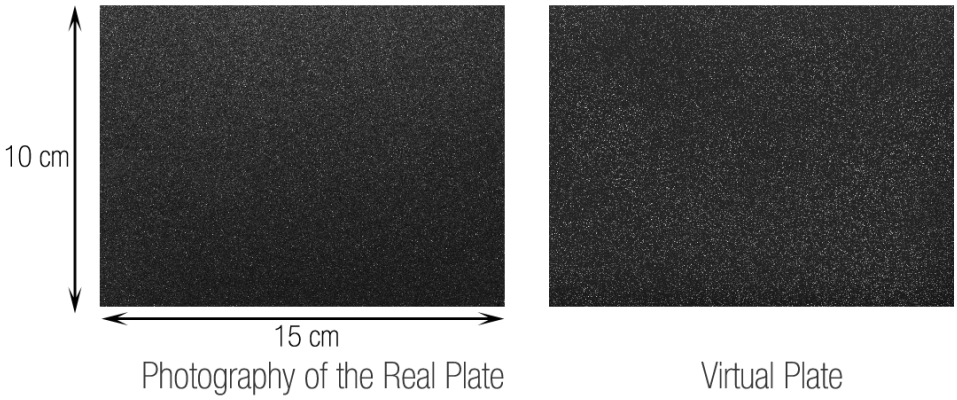


Figure 4: The stimulus displayed to the subjects. The "Reference" material (left plate) is static through the experiment, while the material of the plate "Sample" (right plate) changes according to 13 different values of flake densities, and flake radius size.

a resolution of 1920×1080 pixels at 72Hz. The stimulus consisted of two plates designated as reference, and variable plate, see figure 4. The reference is the photography of the actual automotive paint plate, which was taken with a camera *NikonD*800. The initial high dynamic range (HDR) image was made by taking 6 images with film sensitivity ISO 100, and the following shutter speeds: 1.3, 1.6, 2.0, 2.5, 3.0, and 4.0 seconds. Then, the HDR image image was converted to the RGB colour space using the calibration matrices of the photographic camera, and the screen display. The plates dimensions are 10cm $\times$ 15cm. The samples are placed inside of a dark room with one source of light SOLUX 4700*K* with known spectrum. The virtual stereo and cyclopean cameras are located at 77cm from the plate, which is the same distance of observation that was used to take the photography of the real plate.

5 RESULTS

5.1 Data Exploration

From the box plots, we noticed that there are some extreme outliers and extreme values. This is due to the fact that there is an important agreement within the population to give the same evaluation to a given plate.This is translated into a positive Kurtosis. A principal component analysis (PCA) was applied in order to verify if the observers generally agree in their evaluations. The first axis represents more than 80% in the two studies that is to say that there is an extraordinary agreement between the subjects. Thanks to these preliminary analyses, the average of the panel represents well the raw data so this indicator can be used in Data Modeling.

5.2 Data Modeling

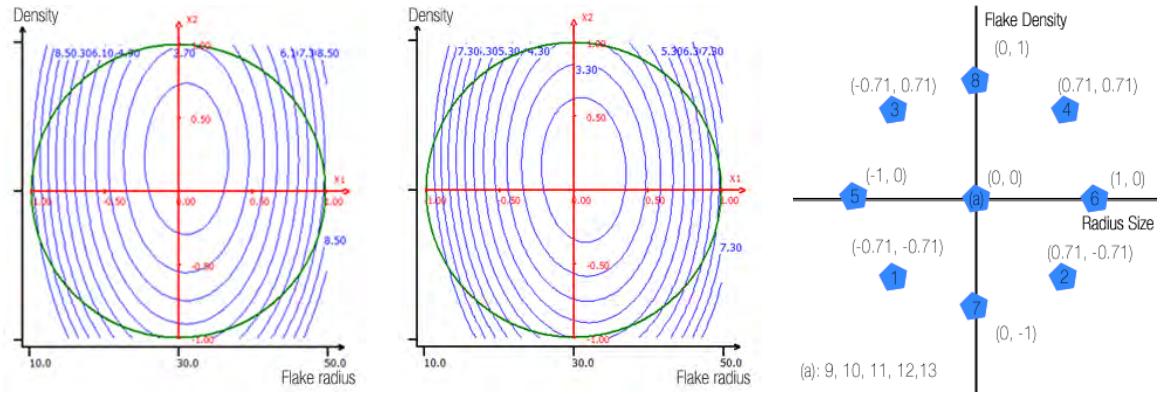
The experimental design allows to evaluate the interaction between the two parameters, non linear effects, and to find the optimum values for each parameter. As we said in the previous section, we can consider that the average of response is representative of the population. Therefore, we can model the response according to the parameters of the experimental design: linear and non linear effects of density and radius size, and their interaction. The surface of response depicted in figures 5 (a) and (b) have a bell shape surface. Hence, there is no interaction between the two factors: flake density and flake radius size. For the stereoscopic case we obtain a better R2 pred (based on cross validation) and a smaller PRESS (Predicted Residual Sum of Squares), see table 3, therefore the stereoscopic model is more robust. The average degree of proximity to the photography of the actual sample is 5.1, and 4.6, for monoscopic and stereoscopic. The analysis of the quadratic effects show that for low or high density of flakes, the virtual plate is not considered similar to the real plate. The same result was found for small and large flakes radius. Finally, the Least Square Difference Test (LSD) show that with the stereoscopic condition the observers were able to discriminate better the plates.

6 DISCUSSION

There is a difference between the the monoscopic and stereoscopic observations of the plate. In average, the stereoscopic images of the virtual plate were better evaluated. The ANOVA repeated measure analysis on the global response indicates that the stereoscopic visualization of the virtual plate is closer to the actual plate. In other words, there is a better visual agreement to the photography of the physical plate with stereoscopic vision. Furthermore, the second ANOVA analysis on stereoscopic and monoscopic condition, confirms that the stereoscopic vision allows a better differentiation of the virtual plates. With monoscopic vision it is more perceptible the individual white reflections of the metallic flakes, and also the visual patterns resulted from the clustering of metallic flakes. While with the

	Coefficient		Signif. %	
	Monoscopic	Stereoscopic	Monoscopic	Stereoscopic
$b0$	2.723	2.423	$< 0.01^{***}$	$< 0.01^{***}$
Radius Size $b1$	-0.590	-0.730	$< 0.0927^{***}$	$< 0.0857^{***}$
Radius Size $b1-1$	6.003	5.323	$< 0.001^{***}$	$< 0.001^{***}$
Flake Density $b2$	-0.794	-0.516	$< 0.0294^{***}$	0.318^{**}
Flake Density $b2-2$	1.810	1.822	$< 0.001^{***}$	0.0123^{***}

Table 2: The table of coefficients.



(a) (b) (c)

Figure 5: The graphs (a) and (b) depict the response surface of the experiment. (c) is the composite design used in the experiment that shows how many different plates to use and the variation of density and size of the metallic flakes. The values depicted in the graph of surface of responses are reduced and centered to the interval -1 to 1 so the influence of the factors can be compared. In this way, we can represent the domain of each factor in the same reference.

	Monoscopic	Stereoscopic
R2	0.961	0.976
R2 pred	0.726	0.838
PRESS	19.494	9.116

Table 3: The adequacy measures for the model.

stereoscopic vision it is more difficult to identify these patterns effects due to the binocular rivalry. Instead we can observe a glittering effect, which according to the comments of the observers, make the virtual plates to look more realistic.

In our experiments, we found that the radius of the flake has a great impact in the judgment of similarity. In figures 5 (a) and (b), the ellipsoidal shape of the isometric curve is oriented in the $x2$ axis. This particular orientation shows that the quadratic effects are stronger in the $x1$ axis, i.e., the flakes radius size axis. This dissymmetry is an indicator that the similarity evaluation note is more sensible to the radius size than to the flake density factor. In addition, according to the table of coefficients 2, the flake radius, $b1-1$, is three times more important than the flake density, $b2-2$. In the stereoscopic case, the shape of the isometric curve is tilted to the right. However, the radius size factor remains more sensible or influent than the flake density parameter.

From the obtained results, there are not particular tuples of factors that makes the subject to judge that the virtual plate is similar to the photography, for instance a tuple with a large flake radius, and a strong flake density. For the monoscopic condition, the optimum values calculated from the surface response, show that in order the virtual plate to be considered similar to the photography, the plate must contain flakes with smaller radius size and to have a higher flake density. While in the stereoscopic condition, the optimum values show the inverse, higher flake radius size with lower flake density.

7 CONCLUSIONS

In this work, we evaluated the pertinence of a "virtual material workshop" approach, and the role of stereoscopy on perception of materials that depict binocular differences such as automotive paints with metallic flakes. For this purpose, we developed a user study based on a design of experiments, to evaluate the visual agreement between the observation of a computer generated object and the actual object. The results show that there is a match between the real and virtual metrics. This means that for a desired appearance our methodology can predict the microstructure. Secondly, the stereoscopic vision improves the visual representation of the virtual plates with metallic flakes. Finally, the size of flake radius has a great influence in the judgment of the observers.

We are currently working on the visualisation of virtual materials with a High Dynamic Range display, to study the influence of the high dynamic luminance on the perception of materials such as car paints. The next step for evaluating VR is to assess the pertinence of using head tracking to generate correct dynamic perspectives.

8 ACKNOWLEDGMENTS

This work is part of the LIMA project (Light Interaction Material Aspect), which is a collaborative research project between academic researchers and industrial manufacturers. The project is funded by the French National Research Agency, ANR, with the Grant Number ANR-11-RMNP-0014. The authors wish to thank F. Willot, E. Couka, D. Jeulin, and P. Callet for the microstructure models, A. Thorel, A. Chesnaud, and M. Ben Achour for the SEM measures, P. Porral for providing PSA Peugeot Citroën paint samples, and the persons who participated in the user study.

9 REFERENCES

[1] Wilkie, A., Weidlich, A., Magnor, M., and Chalmers, A., "Predictive rendering," in [*ACM SIGGRAPH ASIA 2009 Courses*], *SIGGRAPH ASIA '09*, 12:1–12:428, ACM, New York, NY, USA (2009).

[2] Ulbricht, C., Wilkie, A., and Purgathofer, W., "Verification of physically based rendering algorithms," *Computer Graphics Forum* **25**(2), 237–255 (2006).

[3] Rushmeier, H., [*Computer Graphics Techniques for Capturing and Rendering the Appearance of Aging Materials*], 283–292, Springer US (2009).

[4] Enguerrand Couka, François Willot, D. J., "A mixed boolean and deposit model for modeling of metal flakes in paint layers," *Image Anal Stereol* **32**(2), 8–13 (2012).

[5] Nicodemus, F. E., Richmond, J. C., Hsia, J. J., Ginsberg, I. W., and Limperis, T., "Radiometry," ch. Geometrical Considerations and Nomenclature for Reflectance, 94–145, Jones and Bartlett Publishers, Inc., USA (1992).

[6] Boher, P., Leroux, T., and Bignon, T. *SID Conference Record of the International Display Research Conference* .

[7] Kook Seo, M., Yeon Kim, K., Bong Kim, D., and Lee, K. H., "Efficient representation of bidirectional reflectance distribution functions for metallic paints considering manufacturing parameters," *Optical Engineering* **50**(1), 013603–013603–12 (2011).

[8] Jensen, H. W., Marschner, S. R., Levoy, M., and Hanrahan, P., "A practical model for subsurface light transport," in [*Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*], *SIGGRAPH '01*, 511–518, ACM, New York, NY, USA (2001).

[9] Ďurikovič, R. and Martens, W. L., "Simulation of sparkling and depth effect in paints," in [*Proceedings of the 19th spring conference on Computer graphics*], *SCCG '03*, 193–198, ACM, New York, NY, USA (2003).

[10] Ershov, S., Ďurikovič;, R., Kolchin, K., and Myszkowski, K., "Reverse engineering approach to appearance-based design of metallic and pearlescent paints," *Vis. Comput.* **20**, 586–600 (Nov. 2004).

[11] Günther, J., Chen, T., Goesele, M., Wald, I., and Seidel, H.-P., "Efficient acquisition and realistic rendering of car paint," in [*Vision, Modelling, and Visualization 2005 (VMV), Proceedings, November 16-18, Erlangen, Germany*], Greiner, G., Hornegger, J., Niemann, H., and Stamminger, M., eds., 487–494, Akademische Verlagsgesellschaft Aka GmbH, Berlin (November 2005).

[12] Rump, M., Müller, G., Sarlette, R., Koch, D., and Klein, R., "Photo-realistic rendering of metallic car paint from image-based measurements," *Computer Graphics Forum* **27**(2), 527–536 (2008).

[13] Sung, L., Nadal, M. E., McKnight, M. E., Marx, E., and Dutruc, R., "Effect of aluminum flake orientation on coating appearance.," 1–15 (2001).

[14] LEVELT, W. J. M., "Binocular brightness averaging and contour information," *British Journal of Psychology* **56**(1), 1–13 (1965).

[15] Ged, G., Obein, G., Silvestri, Z., Le Rohellec, J., and ViÃ©not, F., "Recognizing real materials from their glossy appearance," *Journal of Vision* **10**(9) (2010).

[16] Obein, G., Knoblauch, K., and Viénot, F., "A framework for the measurement of visual appearance.," *Commission Internationale de l'Eclairage (CIE)* **4**, 711–720 (Aug. 2006).

[17] Templin, K., Didyk, P., Ritschel, T., Myszkowski, K., and Seidel, H.-P., "Highlight microdisparity for improved gloss depiction," *ACM Trans. Graph.* **31**, 92:1–92:5 (July 2012).

[18] Wendt, G., Faul, F., and Mausfeld, R., "Highlight disparity contributes to the authenticity and strength of perceived glossiness," *Journal of Vision* **8**(1) (2008).

[19] Obein, G., Knoblauch, K., and Viénot, F., "Difference scaling of gloss: nonlinearity, binocularity, and constancy.," *Journal of vision* **4**, 711–720 (Aug. 2004).

[20] Sakano, Y. and Ando, H., "Effects of head mo-

tion and stereo viewing on perceived glossiness," *Journal of Vision* **10**(9), 1–14 (2010).

[21] David C. Knill, J. A. S., "Do humans optimally integrate stereo and texture information for judgments of surface slant?," (2003).

[22] McCamy, C. S., "Observation and measurement of the appearance of metallic materials. part i. macro appearance," *Color Research and Application* **21**(4), 292–304 (1996).

[23] McCamy, C. S., "Observation and measurement of the appearance of metallic materials. part ii. micro appearance," *Color Research and Application* **23**(6), 362–373 (1998).

[24] Fettis, G., [*Automotive Paints and Coatings*], Wiley (2008).

Adaptive Importance Caching for Many-Light Rendering

Hiroshi Yoshida
Wakayama University
Wakayama, Japan
s141064@center.wakayama-u.ac.jp

Kosuke Nabata
Wakayama University
Wakayama, Japan
s141044@center.wakayama-u.ac.jp

Kei Iwasaki
Wakayama University,
UEI Research
Wakayama, Japan
iwasaki@sys.wakayama-u.ac.jp

Yoshinori Dobashi
Hokkaido University,
JST CREST
Sapporo, Japan
doba@ime.ist.hokudai.ac.jp

Tomoyuki Nishita
UEI Research, Hiroshima
Shudo University
Hiroshima, Japan
nishita@shudo-u.ac.jp

ABSTRACT

Importance sampling of virtual point lights (VPLs) is an efficient method for computing global illumination. The key to importance sampling is to construct the probability function, which is used to sample the VPLs, such that it is proportional to the distribution of contributions from all the VPLs. Importance caching records the contributions of all the VPLs at sparsely distributed cache points on the surfaces and the probability function is calculated by interpolating the cached data. Importance caching, however, distributes cache points randomly, which makes it difficult to obtain probability functions proportional to the contributions of VPLs where the variation in the VPL contribution at nearby cache points is large. This paper proposes an adaptive cache insertion method for VPL sampling. Our method exploits the spatial and directional correlations of shading points and surface normals to enhance the proportionality. The method detects cache points that have large variations in their contribution from VPLs and inserts additional cache points with a small overhead. In equal-time comparisons including cache point generation and rendering, we demonstrate that the images rendered with our method are less noisy compared to importance caching.

Keywords

Global Illumination, Many-Light Rendering, Importance Sampling, Importance Caching

1 INTRODUCTION

Photorealistic rendering has, for many years, been an interesting and challenging topic in the field of computer graphics. It has been widely used in many applications such as movies, games, architectural design, and so on. Indirect illumination plays an important role in enhancing realism. However, efficient rendering with indirect illumination is still a challenging problem due to the high computational cost.

To compute indirect illumination efficiently, Keller introduced an instant radiosity, which approximates the indirect illumination with virtual point lights (VPLs) [KELLER97]. Many-light rendering [DACHSBACHER14], which extends the instant radiosity, has been extensively researched. Many-light rendering approximates both the direct and indirect illumination incident onto each point to be shaded (referred to as *shading points*) with VPLs. Increasing the number of VPLs increases the accuracy of many-light rendering, but at the cost of computational time.

To handle a large number of VPLs efficiently, importance sampling methods [WANG09, GEORGIEV12, WU13] for VPLs that estimate the outgoing radiance of shading points have been proposed. The key component for the importance sampling method is to construct a probability function that is as proportional as possible to the distribution of contributions from all the VPLs. However, constructing a probability function perfectly proportional to the distribution at each shading point is computationally expensive since it requires a large number of visibility tests between the shading point and all VPLs. Importance caching [GEORGIEV12] constructs a probability function by sparsely distributing the cache points on the surfaces of the scene, and recording the contributions of all VPLs. The probability function at each shading point is calculated by in-

terpolating those at nearby cache points. This method, however, distributes cache points randomly and does not consider the variation in contributions from each VPL. This makes it difficult to construct a probability function that is proportional to the distribution of the VPL contributions, leading to an increase in variance.

To address this problem, we propose an adaptive cache insertion method for a many-light rendering framework. Our method detects regions where the distribution of the VPL contributions varies drastically due to the spatial variations of the shading points, the directional variations of the normals to the shading points, and due to the occlusions between the VPLs and the shading points. Additional cache points are inserted into such regions. In addition, while importance caching calculates the interpolated probability function by simple averaging, our method takes into account the spatial correlation between the shading points and the cache points, and the directional correlation between the normals to the shading points and cache points, and uses these to weight the interpolation. Our results demonstrate that, in an equal time comparison, our method provides better performance (i.e. less variance) than importance caching.

2 PREVIOUS WORK

Many-light rendering, which is based on the instant radiosity as proposed by Keller [KELLER97], distributes a large number of VPLs in the scene and approximates the incident radiance from the direct and indirect illumination from the VPLs. Increasing the number of VPLs increases the rendering accuracy at the cost of computational time. Since several thousand VPLs are required to obtain plausible results, several methods have been proposed that can handle many VPLs efficiently.

To handle a large number of VPLs efficiently, several methods that cluster VPLs have been proposed. Walter et al. proposed a hierarchical representation of the VPLs called Lightcuts [WALTER05]. Hasan et al. proposed the matrix row-column sampling (MRCS) method that samples a small number of VPLs that give a good approximation to the contributions from all the VPLs [HASAN07]. Ou et al. proposed the Lightslice method, which extends the MRCS method by clustering the shading points and applying the MRCS method to each cluster to improve the accuracy [OU11]. Although these clustering methods can efficiently render realistic images by approximating the contributions of the VPLs in each cluster by a representative VPL from each cluster, the rendered images suffer from errors due to VPL clustering.

Importance sampling methods for VPLs have also been proposed. The contribution of a VPL is the product of the incident radiance, the bidirectional reflectance distribution function (BRDF), a geometry term, and the visibility function. By constructing probability functions proportional to the contribution from VPLs and sampling the VPL according to this probability function, the outgoing radiance can be estimated with high accuracy and small variance. Wang and Akerlund proposed a bidirectional importance sampling method for many-light rendering [WANG09]. This method, however, does not take into account the visibility function, resulting in high variance where the incident light is occluded. Wu et al. proposed the VisibilityCluster algorithm, which clusters shading points and the VPLs [WU13]. The visibility function is approximated by the average values of the estimated visibilities between the clusters of shading points and the VPLs. Although this method can render realistic images efficiently, it can fail to sample VPLs with large contributions since estimates of the average values of the visibilities are done by random sampling.

Cache-based methods that exploit correlation to increase the rendering efficiency have been proposed. Ward et al. proposed irradiance caching, which accelerates the indirect illumination calculation by interpolating the incident illumination stored at cache points [WARD88]. Radiance caching methods [KRIVANEK05, KRIVANEK06] store radiance instead of irradiance to efficiently render glossy materials. Visibility caching stores visibility information to accelerate the direct illumination computation [CLARBERG08] . The work that is most relevant to our method is importance caching [GEORGIEV12]. This method randomly distributes cache points, called importance records, in the scene and records the contributions of all the VPLs as shown in Fig. 1(a). Then a probability function perfectly proportional to the contributions of the VPLs at each cache points is calculated. At each shading point, the probability function is calculated by interpolating the contributions stored at nearby cache points, and a small number of VPLs are sampled to estimate the outgoing radiance. Although it can render plausible images efficiently, this method has several drawbacks. Firstly, cache points are distributed randomly. If the VPL contributions stored at the cache points vary drastically, the interpolated probability function may not be proportional to the contributions of the VPLs. Secondly, the interpolated probability function is simply an average of those recorded at nearby cache points, which does not account for the correlation between the shading and cache points. To address this problem, we propose an adaptive cache insertion method for many-light rendering. Our method distributes cache points taking into account variations in the VPL contributions. In addition, our method interpolates the probability function by weighted averages of the probability functions stored at nearby cache points taking into account the correlations between the shading and cache points.

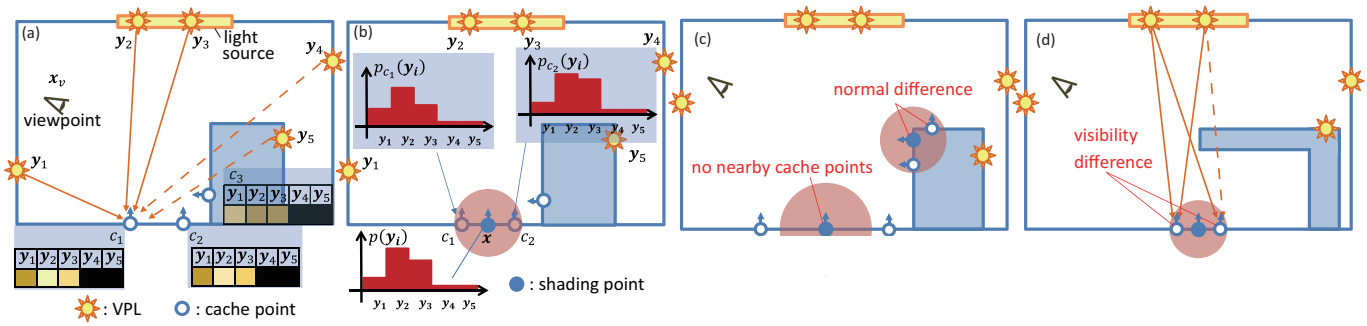


Figure 1: Importance caching (a) records contributions of all VPLs at cache points c_j. Tables under cache points represent VPL contributions. (b) Probability function p at shading point is calculated by averaging those at nearby cache points. Each graph shows the probability function, and the probability functions at cache points are calculated by normalizing VPL contributions. (c) Inefficient cases of importance caching, lack of nearby cache points and lack of cache points with similar normals. (d) Contributions of VPLs at nearby cache points differ due to occlusions.

3 IMPORTANCE CACHING

Importance caching [GEORGIEV12] samples VPLs based on a probability function calculated by interpolation between those stored at cache points. The outgoing radiance $L_o(\mathbf{x},\mathbf{x}_v)$ at shading point $\mathbf{x}$ towards the viewpoint $\mathbf{x}_v$ is estimated by the following equation:

$$L_o(\mathbf{x},\mathbf{x}_v) = \frac{1}{N}\sum_{n=1}^{N} \frac{L(\mathbf{y}_n,\mathbf{x}) f_r(\mathbf{y}_n,\mathbf{x},\mathbf{x}_v) G(\mathbf{x},\mathbf{y}_n) V(\mathbf{x},\mathbf{y}_n)}{p(\mathbf{y}_n)}, \tag{1}$$

where N is the number of sampled VPLs, $\mathbf{y}_n$ is the n-th VPL, and L, f_r, G, and V are the radiance, BRDF, the geometry, and the visibility terms, respectively (please refer to the many-light rendering survey paper [DACHSBACHER14] for more details). The contribution of the VPL is the product of L, f_r, G, and V. The probability function p for sampling the VPLs is expected to be proportional to the distributions of the VPL contributions. However, constructing a probability function perfectly proportional to the distribution of the VPL contributions is computationally expensive since it requires evaluation of all the VPL contributions.

To address this problem, importance caching randomly distributes a small number of cache points in the scene. At each cache point, the contributions from all the VPLs are calculated. The probability function that is perfectly proportional to the distribution of the VPL contributions is calculated by normalizing the distribution. The contribution from the VPLs to the shading point seems to be correlated with those stored at nearby cache points. By exploiting the correlation of the contributions, the probability function p at each shading point is obtained by interpolating those at nearby cache points. However, when geometrical information (e.g. the normal) or the VPL contribution between a shading point and a cache point has a small correlation as shown in Figs. 1(c)(d), the proportionality of the interpolated probability function decreases.

4 PROPOSED METHOD

Instead of random sampling, our method distributes cache points taking into account the geometrical information of the shading points and the distribution of the VPL contributions. Fig. 2 shows an overview of our method.

4.1 Generating Initial Cache Points

The contributions from a VPL to two shading points $\mathbf{x}_i, \mathbf{x}_j$ have large correlation when the positions $\mathbf{x}_i, \mathbf{x}_j$ and the normals $\mathbf{n}_i, \mathbf{n}_j$ to the shading points are similar. By exploiting this, our method first clusters the shading points based on the positions and the normals, employing the clustering method described in [OU11]. The shading points are represented by 6-dimensional points consisting of the positions and the normals. Firstly, the positions of the shading points are normalized into $[-1,1]^3$, which is equal to the range of the normals. The bounding box of the 6-dimensional points is calculated, and then recursively subdivided until the number of 6-dimensional points or the size of bounding box is smaller than the thresholds. The bounding box is split along its longest axis. After the subdivision is terminated, one shading point is randomly sampled from each cluster and is used as the cache point. At each cache point, the contributions from all the VPLs are calculated and a cumulative distribution function is constructed.

4.2 Adaptive Insertion of Cache Points

The initial cache points are distributed according to the similarity of the shading points, but not considering the contributions of VPLs. For example, as shown in Fig. 1(d), the contributions of VPLs can differ at nearby cache points due to occlusions, leading to the interpolated probability function having reduced proportionality and increased variance.

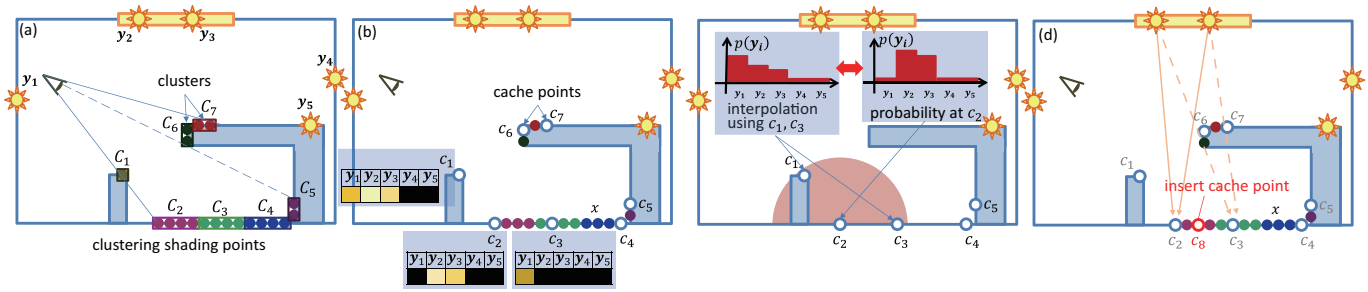


Figure 2: Overview of our method. (a) Clustering shading points based on their positions and normals. (b) Calculate contributions of VPLs at each cache point c_j. (c) Calculate sum of differences of p at c_2 and interpolated probability function using c_1 and c_3. (d) Insert new cache point c_8 from cluster C_2.

To address this problem, our method exploits the fact that each cache point records the contributions of all VPLs and therefore, an ideal probability function perfectly proportional to the contributions of all the VPLs is easily obtained. Our method detects those nearby cache points whose recorded contributions differ due to occlusions, and inserts additional cache points for such regions. If the VPL contributions recorded at cache points near to c_j differ drastically from those at c_j due to occlusions, the interpolated probability function differs from the probability function of c_j. Therefore, our method calculates the sum of the differences between the probability function recorded at c_j and interpolated probability function from nearby cache points of c_j. If the sum of differences exceeds the threshold δ, an additional cache point is inserted. The threshold δ is set experimentally in the current implementation.

The VPL contributions at the additional cache point need to have large correlation with those recorded at c_j. To correlate VPL contributions between the additional cache point and c_j, small variations in the geometrical information and the occlusions are required. However, computing the visibilities between all VPLs and a cache point is computationally expensive, it is difficult to detect the variations in the occlusions with a small overhead. Our method calculates the positions of the additional cache points using the geometrical information of c_j. Since the shading points in the cluster C_j corresponding to c_j have similar geometry information, our method samples one shading point randomly from C_j. The insertion process for all the cache points is repeated until the number of inserted cache points is smaller than a threshold. The cache points are stored in a kd-tree for fast search of cache points near to each shading point.

Fig. 3 shows the initial cache points (left) and the adaptively inserted cache points (right) of a Cornell box scene. The initial cache points are distributed uniformly on the surfaces of the scene, while the inserted cache points are distributed near the boundaries of shadows, where the visibilities between the VPLs and the cache points change.

4.3 Rendering

The outgoing radiance $L_o(\mathbf{x}, \mathbf{x}_v)$ at shading point $\mathbf{x}$ is calculated by sampling VPLs according to the probability function p interpolated from those recorded at a number, M, of nearby cache points. In contrast to the simple average as in importance caching [GEORGIEV12], our method calculates the probability function p using a weighted average that considers the spatial and directional correlations between the shading and cache points. The probability function p is calculated by the following equation:

$$p(\mathbf{y}_n) = \sum_{k=1}^{M} w_k p_k(\mathbf{y}_n), \tag{2}$$

where M is the number of cache points. $M = 3$ works well for our method as proposed in [GEORGIEV12]. w_k and p_k are the weight and probability function for the k-th nearest cache point, respectively. The weight w_k is calculated using the formula proposed in [CLARBERG08].

$$w_k = \sqrt{1 - |\mathbf{n} \cdot \mathbf{v}|} \cdot \hat{w}(d, \theta), \tag{3}$$

where $\mathbf{n}$ is the normal to shading point $\mathbf{x}$, and $\mathbf{v}$ is the normalized vector from $\mathbf{x}$ to the k-th nearest cache point c_k. d is the distance between the shading point and the cache point, and θ is the angle between the normals to the shading and cache points. The weight function $\hat{w}$ is calculated from the unnormalized weight function w:

$$w(d, \theta) = \left(1 - \frac{\theta}{\pi}\right)\left(1 - \frac{d'}{1 + \lambda d'}\right), \tag{4}$$

where $d' = d/d_{max}$, d_{max} is the maximum search range, and λ is a parameter. The weight function $\hat{w}$ is calculated from $\hat{w} = (w(d, \theta) - w(d_{max}, \theta_{max}))/(1 - w(d_{max}, \theta_{max}))$, where θ_{max} is the maximum angle between the normals. $\theta_{max} = \pi/6$ and $\lambda = 5$ are used as proposed in [CLARBERG08].

5 RESULTS

Figs. 4, 5, and 6 show equal-time comparisons between our method and importance caching [GEORGIEV12].

Figure 3: Example of cache points (red points) in the Cornell box scene. Left: initial cache points generated by sampling each cluster of shading points. Right: inserted cache points generated by considering VPL contributions.

The computational times are measured on a desktop PC with an Intel Xeon CPU E3 1270 3.4GHz. All the computations were performed in parallel using multi-threading. The image resolutions are 1024×768. All the computations used in our method, except for cache generation, adaptive insertion, and weight calculation, are the same as those used in importance caching. Cache points are generated randomly in importance caching, and the same numbers of cache points are used in our method as in importance caching. As described in [GEORGIEV12], four sampling strategies are used. Bilateral multiple importance sampling using the α-max heuristic is performed for both methods. Our method distributes VPLs in the same way as described in [DACHSBACHER14]. The reference images are rendered by accumulating the contributions from all the VPLs. Table 1 shows the statistics of our results.

Fig. 4 shows an equal-time comparison of a Sibenik scene. The computational times (cache generation/rendering) for our method and importance caching were 36.6s (17.8s for cache generation/18.8s for rendering) and 33.7s (15.2s/18.5s), respectively. Fig. 4(a) shows the result rendered using our method. Figs. 4(b), (c), and (d) show close-ups of the area outlined in red in Fig. 4(a) for the reference image, and the results rendered by our method, and importance caching, respectively. Figs. 4(e), (f), and (g) show close-ups of the area outlined in blue in Fig. 4(a). As shown in these images, our method can render images with less noise compared to importance caching. Importance caching [GEORGIEV12] tends to distribute cache points near the viewpoint. Therefore regions far from the viewpoint (e.g. the windows in Fig. 4(d)) have less cache points, resulting in noisy images. In addition, since importance caching does not take into account occlusions in distributing cache points, the regions where occlusions vary drastically (e.g. Fig. 4(g)) suffer from noise, whereas our method can distribute cache points for such regions, resulting in less noise as shown in Fig. 4(f).

Table 1: Statistics of results. N_T, N, and N_c are the number of triangles, VPLs, and cache points, respectively.

Scene	N_T	N	N_c
Sibenik (Fig. 4)	75,284	7,785	2,950
Sponza (Fig, 5)	66,450	6,479	1,728
Conference (Fig. 6)	331,179	5,133	2,478

Fig. 5 shows an equal-time comparison of a Sponza scene. As shown in Figs. 5(b) to (g), our method can render less noisy images especially for regions (e.g. arches and pillars) where the visibilities between the VPLs and the shading points change. The computational times (cache generation/rendering) for our method and importance caching were 24.7s (7.5s/17.2s) and 23.3s (5.9s/17.3s), respectively.

Fig. 6 shows an equal-time comparison of a Conference scene. Figs. 6(b)(d), (c)(f), and (d)(g) show close-ups of the reference image, the results rendered by our method, and importance caching, respectively. In Fig. 6(d), a large variance due to the occlusion due to the table appears in the chair, whereas our method (Fig. 6(c)) renders an image comparable to the reference image shown in Fig. 6(b). The computational times for our method and importance caching were 18.7s (6.5s/12.2s) and 18.2s (6.4s/11.9s), respectively.

Figs. 7, 8, and 9 show visualizations of the root-mean-square-error (RMSE) between each method and reference images rendered by summing all the VPL contributions. The color bar shows the false color. For the Sibenik scene, the RMSE for our method is 0.0486773 while that for importance caching is 0.0662813. As shown in Fig. 7, our method can render less noisy images especially near windows and pillars. In the Sponza scene, the RMSE for our method is 0.146164 while that of importance caching is 0.207015. Since it is difficult for importance caching to distribute cache points inside the scene, large variance can appear as shown in Fig. 8(b), while our method can lessen this as shown in Fig. 8(a). In the Conference scene (Fig. 9), the RMSE for our method and importance caching are 0.0294903 and 0.032524, respectively. As shown in Fig. 9, by inserting additional cache points, our method reduces the variance near chairs occluded by the table.

Since, with our method, new cache points are added in regions where the variations in the VPL contributions are large, for the same number of cache points, the cache points are distributed more sparsely in other regions compared to importance caching, resulting in a slightly increased variance (e.g. near the floor in Fig. 8). However, our method can reduce the RMSE for the overall scene as shown in Figs. 7, 8, and 9.

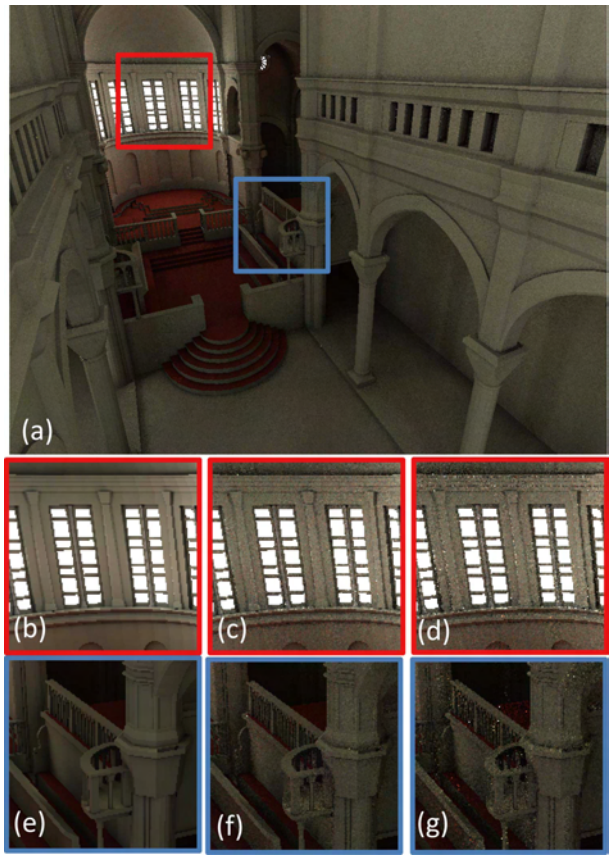


Figure 4: Sibenik scene. (a) rendering result of our method. (b)(c)(d) close-up images of reference, our method, and importance caching, respectively. (e)(f)(g) close-up images of reference, our method, and importance caching. Our method can render less noisy image in equal time rendering compared to importance caching.

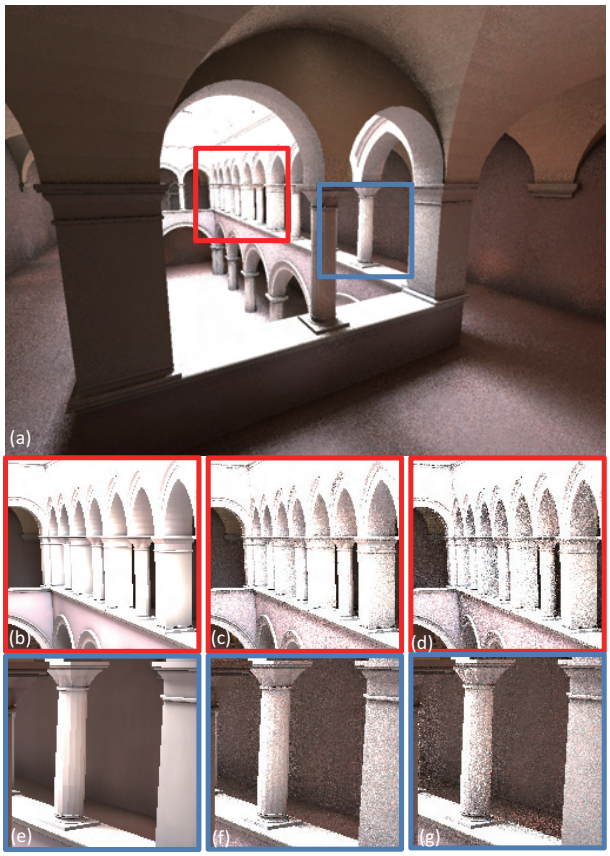


Figure 5: Sponza scene. (a) rendering result of our method. (b)(c)(d) close-up images of reference, our method, and importance caching, respectively. (e)(f)(g) close-up images of reference, our method, and importance caching. Our method can render less noisy images, especially near arches and pillars in equal time rendering.

To inspect the effectiveness of the adaptive insertion of cache points and the weighting function that considers the spatial and directional correlations, our method renders the Sibenik scene using adaptively inserted cache points and uniform weights used in [GEORGIEV12]. The RMSE in this case is 0.0516844, while that with random cache points and uniform weights is 0.0662813. As shown in this experiment, adaptive cache insertion contributes to the improvements most.

6 CONCLUSIONS AND FUTURE WORK

We have proposed an adaptive cache insertion method for importance caching. Our method clusters the shading points and selects cache points from clusters and exploits the spatial and directional correlations between shading points and cache points. Our method detects cache points whose VPL contributions differ from those of nearby cache points and inserts further cache points, resulting in reduced variance compared to that obtained in equal-time rendering using the original importance caching method.

For future work, we plan to accelerate our method using VPL clustering. Moreover, we propose to distribute cache points taking into account the scene saliency.

ACKNOWLEDGEMENTS

This research was partially supported by JST, CREST.

7 REFERENCES

[KELLER97] A. Keller, Instant Radiosity, Proc. of SIGGRAPH '97, pp. 49-56, 1997.

[DACHSBACHER14] C. Dachsbacher, J. Krivanek, M. Hasan, A. Arbree, B. Walter, J. Novak, Scalable Realistic Rendering with Many-Light Methods, Computer Graphics Forum, Vol. 33, No. 1, pp. 88-104, 2014.

[WANG09] R. Wang, O. Akerlund, Bidirectional Importance Sampling for Unstructured Direct Illumination, Computer Graphics Forum, Vol. 28, No. 2, pp. 269-278, 2009.

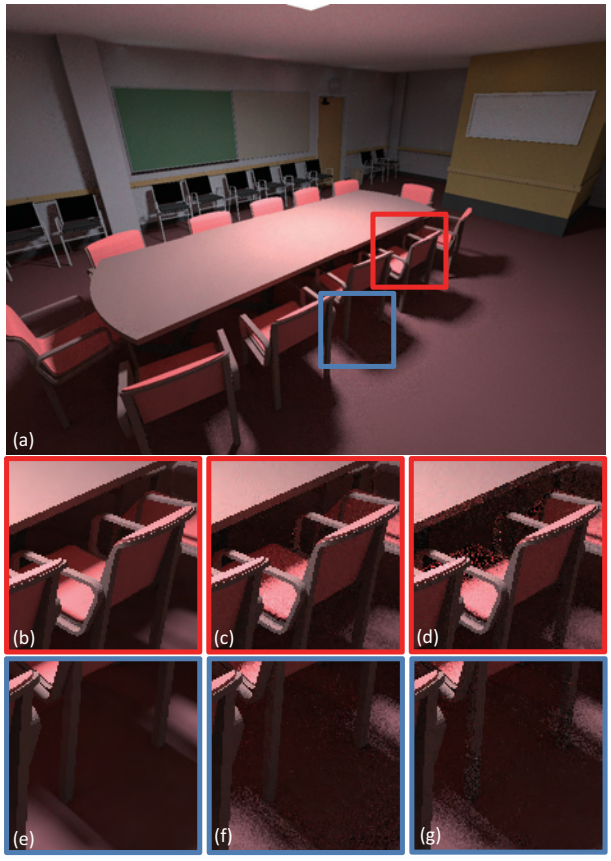


Figure 6: Conference scene. (a) rendering result of our method. (b)(c)(d) close-up images of reference, our method, and importance caching, respectively. (e)(f)(g) close-up images of reference, our method, and importance caching. Our method can render less noise images, especially for chairs in equal time rendering.

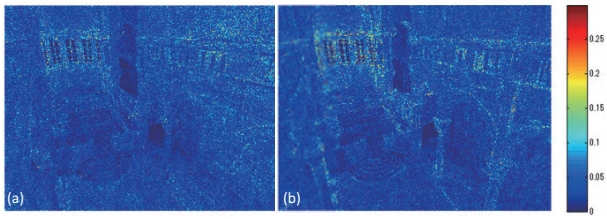


Figure 7: Error images of Sibenik scene. (a) our method and (b) importance caching. RMSEs of our method and importance caching are 0.0486773 and 0.0662813, respectively.

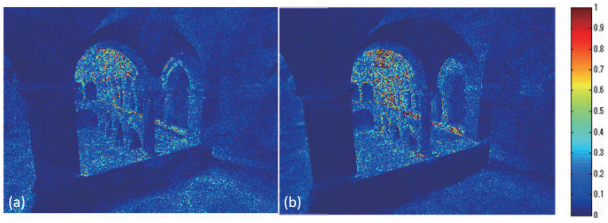


Figure 8: Error images of Sponza scene. (a) our method and (b) importance caching. RMSEs of our method and importance caching are 0.146164 and 0.207015, respectively.

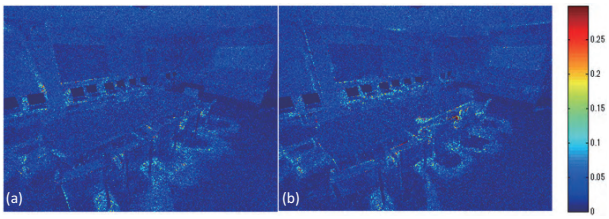


Figure 9: Error images of Conference scene. (a) our method and (b) importance caching. RMSEs of our method and importance caching are 0.0294903 and 0.032524, respectively.

[WU13] Y. T. Wu, Y. Y. Chuang, VisibilityCluster: Average Directional Visibility for Many-Light Rendering, IEEE Transactions on Visualization and Computer Graphics, Vol. 19, No. 9, pp. 1566-1578, 2013.

[GEORGIEV12] I. Georgiev, J. Krivanek, S. Popov, P. Slusallek, Importance Caching for Complex Illumination, Computer Graphics Forum, Vol. 31, No. 2, pp. 701-710, 2012.

[WALTER05] B. Walter, S. Fernandez, A. Arbree, K. Bala, M. Donikian, D. Greenberg, Lightcuts: A Scalable Approach to Illumination, ACM Transactions on Graphics, Vol. 24, No. 3, pp. 1098-1107, 2005.

[HASAN07] M. Hasan, F. Pellacini, K. Bala, Matrix Row-Column Sampling for the Many-Light Problem, ACM Transactions on Graphics, Vol. 26, No. 3, pp. 26:1-26:10, 2007.

[OU11] J. Ou, F. Pellacini, LightSlice: Matrix Slice Sampling for the Many-Lights Problem, ACM Transactions on Graphics, Vol. 30, No. 6, pp. 179:1-179:8, 2011.

[CLARBERG08] P. Clarberg, T. Akenine-Moller, Exploiting Visibility Correlation in Direct Illumination, Computer Graphics Forum, Vol. 27, No. 4, pp. 1125-1136, 2008.

[WARD88] G.J. Ward, F.M. Rubinstein, R.D. Clear, A Ray Tracing Solution for Diffuse Interreflection, ACM SIGGRAPH Computer Graphics, Vol. 22, No. 4, pp. 85-92, 1988.

[KRIVANEK05] J. Krivanek, P. Gautron, S. Pattanaik, K. Bouatouch, Radiance Caching for Efficient Global Illumination Calculation, IEEE Transactions on Visualization and Computer Graphics, Vol. 11, No. 5, pp. 550-561, 2005.

[KRIVANEK06] J. Krivanek, K. Bouatouch, S. Pattanaik, J. Zara, Making Radiance and Irradiance Caching Practical:Adaptive Caching and Neighbor Clamping, Proc. of Eurographics Symposium on Rendering, pp. 127-138, 2006.

Isosurface Orientation Estimation in Sparse Grids Using Tetrahedral Splines

Brian A. Wood
University of Alabama in Huntsville
Huntsville, Alabama 35899
bwood@cs.uah.edu

Timothy S. Newman
University of Alabama in Huntsville
Huntsville, Alabama 35899
tnewman@cs.uah.edu

ABSTRACT

One challenge in applying standard marching isosurfacing methods to sparse rectilinear grid data is addressed. This challenge, the problem of finding approximating gradients adjacent to locations with data dropouts, is addressed here by a new approach that utilizes a tetrahedral spline fitting-based strategy for gradient approximation. The new approach offers improved robustness in certain scenarios (compared to the current state-of-the-art approach for sparse grid isosurfacing). Comparative studies of the new approach's accuracy and computational performance are also presented.

Keywords

Orientation Estimation, Volume Visualization, Isosurfaces

1 INTRODUCTION

One common means for visualizing scalar volumetric data is isosurfacing, which involves finding the set of locations in space where the phenomenon recorded in the dataset achieves a particular value, called the isovalue, denoted herein as α. Isosurface visualization is a powerful approach for observing and studying the behavior of volumetric data. Isosurfacing can promote discovery in disparate applications areas, such as medical diagnosis, fluid flow studies, etc.

Well-known isosurfacing methods exist for volumetric data organized on a number of grid types [10]. Focus here is on scalar data organized on rectilinear grids, which is very common, and on isosurfacing methods applied to such grids that produce triangle meshes approximating the isosurface and assume data values are available at each grid point. However, in some applications, the data is sparse; there is not a data value available at every grid point. (Here, we will use the term sparse grid to mean a 3D rectilinear grid dataset with some missing values.) For example, data collected from sensor arrays may have missing data values when data cannot be collected at every grid point due to physical limitations. Popular isosurfacing methods for rectilinear grid data, such as the standard, marching methods of Marching Cubes and Marching Tetrahedra, require determination of a local gradient at each mesh vertex to estimate the isosurface orientation, and then use that in rendering to produce a shading that is harmonious with local data trends. When data is sparse, the schemes these methods use for estimating orientation can fail at certain locations. Thus, sparseness can make well-known isosurfacing rendering methods unable to be applied. Here, we introduce a new solution to the challenge of isosurfacing on sparse grids.

Sparse grids may be produced from a variety of sensing modalities and volume data generation methods. Data from sensor arrays, particularly ones that measure physical phenomena, has the potential to have missing data values due to sensor faults. For example, wireless 3D sensor arrays, such as those used to capture data underground [1] and underwater [17], operate under harsh conditions and can be particularly vulnerable to sensor faults. Low batteries, bad calibration, high noise, or environmental hazards can all contribute to faults in sensor arrays [11]. Conversion of 3D mesh geometry to volume data via voxelization algorithms [16] can produce datasets with data values only at grid points necessary to reproduce the original mesh. Additionally, volume data derived from point clouds or signed distance functions may not contain sufficient data to estimate data gradients at all isosurface locations, in particular the mesh vertices [12].

One prior work has proposed a work-around to the gradient (orientation) determination challenge in Marching Cubes on sparse grids. The new approach we describe here offers improved results in certain scenarios.

The paper is organized as follows. Section 2 discusses background material and related work. Section 3 de-

scribes the new approach for estimating isosurface orientation on sparse grid datasets. Section 4 provides details on rendering isosurfaces extracted from sparse grids. Section 5 provides results from experiments and comparisons to prior orientation (or normal) estimation approaches. Section 6 contains the paper's conclusion.

2 BACKGROUND AND RELATED WORK

The most common method [10] for isosurfacing on scalar data on rectilinear grids is the Marching Cubes (MC) algorithm. MC has been adapted by Nielson et al. [12] to allow application to rectilinear grids with missing data values (i.e., sparse grids). We describe that adaptation in Section 3.2. First, though, we describe the basic steps of MC and illustrate its failings for sparse grids.

Marching Cubes isosurfacing produces a triangle mesh representation of the isosurface by advancing cell-by-cell through the volume. In each cell, it follows three major steps. In the first step, the general topological arrangement of the isosurface mesh in the cell is determined. (Each general topological arrangement is called a "case" in this paper, reflecting the typical nomenclature of the MC literature.) Second, for topologies containing isosurface mesh facets, the mesh vertex locations in the cell are found. Third, the triangle mesh is formed by connecting vertex locations into the determined topology. An orientation vector is also determined for each vertex location.

In MC, mesh vertices are located on grid lines, with positions there found via linear interpolation. At each vertex, an orientation vector is ultimately used in rendering the produced mesh. These vectors are determined by linearly interpolating the gradients of the grid point locations bounding the grid segment containing each mesh vertex. These gradients are computed using central differencing; for grid point (x_i, y_i, z_i), MC finds the gradient ∇f as:

$$\nabla f(x_i, y_i, z_i) = \left\{ \begin{array}{c} \dfrac{f(x_{i+1}, y_i, z_i) - f(x_{i-1}, y_i, z_i)}{2} \\ \dfrac{f(x_i, y_{i+1}, z_i) - f(x_i, y_{i-1}, z_i)}{2} \\ \dfrac{f(x_i, y_i, z_{i+1}) - f(x_i, y_i, z_{i-1})}{2} \end{array} \right\}, \tag{1}$$

where $f(x_i, y_i, z_i)$ is the scalar value at (x_i, y_i, z_i).

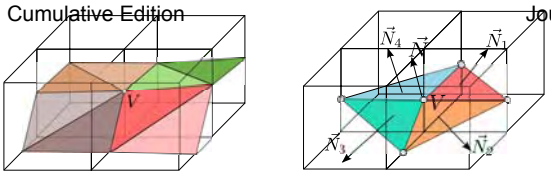
Since the central-difference gradient uses the values of adjacent grid points, if there is a missing data value preceding or following a grid point in any axial direction, central-differencing will be undefined. As a result, MC is unable to estimate the orientation vector for any mesh vertex on a grid segment whose endpoint has an undefined gradient value. Data sets with missing or undefined data thus require an alternative orientation estimator. One option could be use of ad-hoc alternatives for those grid points where central differencing is undefined. For example, a mix of methods could be used (e.g., forward-differencing and reverse-differencing, as suitable) at a cost of consistency.

Other works have considered the issue of estimating orientation in volume data without relying on differencing techniques. For example, Möller et al. [15] have used a two-step approach for shading raytraced isosurface renderings. Hossain et al. [8] have proposed reconstruction filters for gradient estimation derived from methods using Taylor series and Hilbert spaces. They evaluated the accuracy of their filters on both Cartesian and Body-Centered Cubic lattices. Correa et al. [4] have studied averaging-based and regression-based orientation estimation approaches for use in volume raycasting on unstructured grids. Their study recommended the use of a hybrid approach that selects the gradient estimator to use based on local properties of the unstructured grid. Neumann et al. [9] have estimated orientation by fitting a hyperplane on points nearby to a grid point and then taking a linear regression result on data points on the hyperplane. However, while these orientation estimation approaches do not rely on differencing, they assume that data is available at all grid points and thus cannot be used with sparse grids.

Other methods for producing visualizations of sparse grid volume data have also been described. For example, Djurcilov and Pang [6] have described some techniques for visualizing weather data when sample points are missing due to sensor failures. Their techniques require resampling data to produce a fully populated grid prior to isosurface extraction.

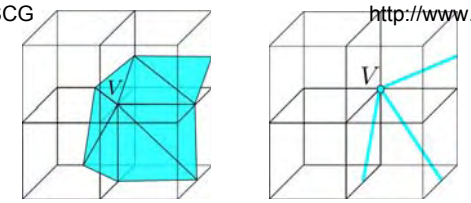
2.1 Quadratic and Quintic Splines

Rössl et al. [14] have proposed a technique for volume reconstruction by fitting a spline model to regular, rectilinear volumetric data. Their technique first partitions the volume's grid into uniform tetrahedra and then fits super splines on each partition. Super splines are a class of splines in which smoothness is preserved on vertices between adjacent tetrahedra. Each fitting uses Bézier splines with constants drawn from the values at the vertices of each tetrahedron, ensuring that the super spline condition is not violated. Details of their process are described later, in Section 3. Awanou and Lai [2] have presented an approach using quintic splines to interpolate a volume. Their approach is similar to that of Rössl et al., but it does not require a regular grid and uses a higher order spline to model the volume data.



(a) Marching Cubes output (b) LSN virtual local mesh

Figure 1: LSN perp vector estimation



(a) Non-degenerate mesh (b) Degenerate mesh used by LSN

Figure 2: LSN summed average estimation

2.2 Locally Supported Normals (LSN)

One methodology for determining isosurface orientation in sparse rectilinear grid data is the Locally Supported Normal (LSN) approach described by Nielson et al. [12]. It considers isosurfacing in a Marching Cubes context, resolving the undefined orientation problem by an estimation process that uses a virtual mesh constructed on the vertices of the MC isosurface. Orientation vectors computed in this manner tend to exhibit sharper shading color transitions at triangle edges than if central-differencing could be used, resulting in a surface with a more faceted appearance. However, central-differencing cannot be applied where grid values are missing or undefined.

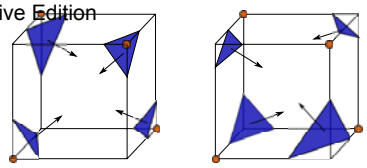
The estimation used in LSN is integrated into MC-style isosurfacing; it produces orientation estimates as vertex locations are calculated. The LSN approach relies on a temporary virtual local mesh that it defines about the point for which an orientation vector is needed. This virtual mesh is *not* the Marching Cubes output mesh; Figure 1 demonstrates the difference between a mesh produced by MC and the virtual mesh used by LSN for a point V. The LSN approach first computes perpendicular (perp) vectors for each face in the virtual mesh; these vertex perp operations are done independent of the MC topology determination. Each perp vector is found as the cross-product of edge vectors of the virtual mesh face. For each of the MC internal vertices shared by multiple triangles, all perp vectors of faces incident to it are averaged to form a *master perp vector* at the vertex. The master perp vector becomes the LSN's estimate of the isosurface orientation at that vertex. Figure 1(b) shows the LSN's estimation of the orientation for a location V in a volume. Four perp vectors, $\vec{N}_1, \vec{N}_2, \vec{N}_3$, and $\vec{N}_4$, are shown. The average of these is the master perp vector $\vec{N}$; here, $\vec{N}$ is $\frac{1}{4}\sum_{i=1}^{4}\vec{N}_i$.

The LSN's estimation can produce erroneous results when certain data characteristics are encountered. The first, and most pronounced, of these errors occurs when degenerate triangles are encountered during orientation estimation. A degenerate triangle with two or more coincident vertices will yield a cross-product of zero, resulting in a zero vector (because the triangle does not lie on a unique plane in space). MC produces degenerate triangles when the isovalue is identical to a grid point value [13]. If a vertex is associated with only degenerate triangles, the orientation vector computed using LSN at that vertex has length zero. The rendered isosurface can contain artifacts at pixel locations affected by the zero length orientation vector.

In Figure 2(a), a mesh containing no degenerate triangles is shown. In contrast, Figure 2(b) displays an LSN mesh corresponding to the same topology, but with vertex V located at a cube corner, resulting in four degenerate triangles (one triangle degenerating to a point and three to a line). The result from LSN is a zero length orientation vector.

Additionally, the LSN approach makes assumptions about what have been called ambiguous faces [10] of cells. These assumptions can lead to inaccurate orientation vectors. One example cube where this incorrect assumption is a problem is shown in Figure 3. The cube has the Case 13 base topology of the MC [12], shown in Figure 3(a). However, the LSN estimation uses the virtual mesh shown in Figure 3(b) to compute orientation vectors in corners of the cube opposite to those defined by MC. We refer to triangles used in the LSN virtual mesh that do not appear in the MC mesh as *illusory triangles*. The normals (i.e., perp vectors) associated with these triangles may differ greatly from the orientation vectors that would result if the actual MC isosurface facets had been used. In particular, each vector found using an illusory triangle will contribute errors to the orientation vector estimation at vertices of illusory triangles. For such situations, the orientations can be estimated incorrectly and yield incorrectly shaded renderings.

The illusory triangle problem in LSN is not just limited to cubes with ambiguous faces. For example, in Marching Cubes Case 5 LSN uses an illusory triangle to compute a perp vector. The topology used by MC for the Case 5 topology is shown at the top of Figure 4 (labeled "C5"). The five virtual mesh triangles used by LSN are shown in the rest of the figure. While most of the virtual mesh triangles should produce reasonable results, the one used for V_4 is illusory and its orientation is not consistent with the actual mesh properties at V_4. Other MC cases also exhibit illusory triangles yielding orientations that differ markedly from that of the MC isosurface mesh. An example of the incorrect orientation from illusory triangles is provided later in this work.



(a) Topology of cell as defined by Marching Cubes

(b) Topology of cell assumed by the LSN approach

Figure 3: Comparison of cell topologies used by MC and LSN

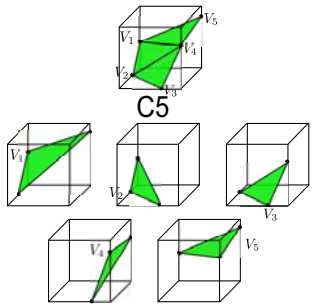


Figure 4: Case 5 MC and LSN topologies

3 NEW GRADIENT ESTIMATION APPROACH FOR SPARSE GRIDS

Next, we describe our new approach for determining orientation vectors in MC for sparse grids. The approach is guaranteed to produce orientation vectors at any location for which it is possible to find a Marching Cubes isosurface vertex. That is, the scheme introduced here can handle any rectilinear sparse grid configuration satisfying the condition that the isosurface vertices can be computed. (I.e., like LSN, our approach assumes there is local support for the isosurface.) For some scenarios, it also offers improved performance over prior approaches for computing MC isosurface orientation vectors on sparse grids.

3.1 Using Quadratic Splines

Our work is motivated by Rössl et al.'s modeling of volumetric data variation using quadratic Bézier-Bernstein super splines (2BBSS) in tetrahedral regions. A tetrahedron allows for the use of an interpolating volumetric spline using a barycentric coordinate system given a sufficient number of data points on the tetrahedron. Specifically, given four points v_0, v_1, v_2, v_3 defining the four vertices of a tetrahedron, a quadratic trivariate spline p is composed in the Bézier-Bernstein form:

$$p(\lambda) = \sum_{i+j+k+l=2} a_{ijkl} B_{ijkl}(\lambda), \qquad (2)$$

where the parameter λ is the location within the spline (in barycentric coordinates with $\lambda = (\lambda_0, \lambda_1, \lambda_2, \lambda_3)$), the coefficients a_{ijkl} are the control points of the spline,

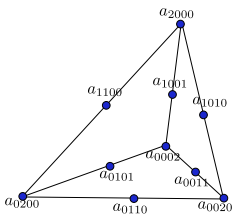


Figure 5: Spline control points

and the B_{ijkl}'s are Bernstein polynomials. The control points are calculated as linear combinations of the vertices of the tetrahedron:

$$a_{ijkl} = \frac{i}{2}v_0 + \frac{j}{2}v_1 + \frac{k}{2}v_2 + \frac{l}{2}v_3, \qquad (3)$$

as depicted in Figure 5. The Bernstein polynomials B_{ijkl} are defined as

$$B_{ijkl}(\lambda) = \frac{2!}{i!j!k!l!}\lambda_0^i\lambda_1^j\lambda_2^k\lambda_3^l,\ i+j+k+l=2, \qquad (4)$$

where each $\lambda = (\lambda_0, \lambda_1, \lambda_2, \lambda_3)$ is a barycentric coordinate with respect to the tetrahedron.

Numerous schemes exist for partitioning rectilinear grids into collections of tetrahedra. We employ one such scheme here to enable the use of tetrahedral splines in the estimation of orientation vectors. Tetrahedral partitions also alleviate the problem of missing data because only 4 grid values are needed to model isosurface behavior within a tetrahedral partition, as opposed to the 6 necessary for a central differencing. By partitioning rectilinear dataset cells into tetrahedra, we can calculate an orientation in any cell intersected by the isosurface. In the 2BBSS model, each tetrahedron must have associated data values at each tetrahedral vertex. Given such, a spline is formulated that approximates the surface within the tetrahedron.

Our approach finds the approximating spline in cells intersected by the isosurface by partitioning the cell into tetrahedra and then evaluating the spline constructed on those tetrahedra to determine orientation vectors (i.e., spline normals) at any barycentric coordinate $(\lambda_0, \lambda_1, \lambda_2, \lambda_3)$ within each tetrahedron of interest. For each of them, our approach uses de Casteljau's algorithm [3] [5] to determine the spline's partial derivative [14] by applying the algorithm in the direction of tetrahedron edges. The usage of de Casteljau's algorithm to compute the derivative of a curve is well understood [7].

For any point on a spline, the formulation of de Casteljau's algorithm enables finding the directional derivatives at q as follows. First, given a spline with control points of the form a^0_{ijkl}, for a point q having the

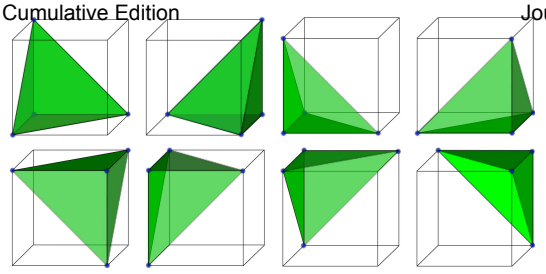
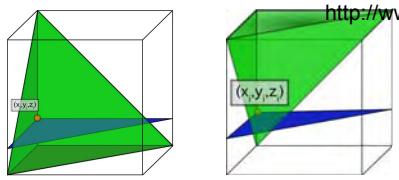


Figure 6: Tetrahedral partitions



(a) Tetrahedral partition 1 (b) Tetrahedral partition 2

Figure 7: Two choices of tetrahedral partition of the cell

barycentric coordinate $\lambda_{\mathbf{q}} = (\lambda_{0_q}, \lambda_{1_q}, \lambda_{2_q}, \lambda_{3_q})$, new control points denoted by a^1_{ijkl} are computed as:

$$a^1_{ijkl} = \lambda_{0_q} a^0_{i+1,j,k,l} + \lambda_{1_q} a^0_{i,j+1,k,l} + \lambda_{2_q} a^0_{i,j,k+1,l} + \lambda_{3_q} a^0_{i,j,k,l+1}, \quad (5)$$

which define a subdivision of the original spline. (Another application of the formula would produce the value at q, however we need just the control points a^1_{ijkl} of the spline subdivision because they define partial derivatives for the spline.) Since the normal at any point on a surface $s(x,y,z)$ can be defined as

$$\nabla s(x,y,z) = (\frac{\partial s}{\partial x}, \frac{\partial s}{\partial y}, \frac{\partial s}{\partial z}), \quad (6)$$

we compute the orientation by finding the partial derivatives in the directions parallel to the coordinate system axes. The formulation of this partial derivative is given in Section 4.

4 ISOSURFACE RENDERING WITH SPARSE GRIDS

Our approach defines 2BBSS splines for tetrahedral subregions of each active cell. We consider eight candidate tetrahedral partitions of each cell (shown in Figure 6) and choose from these the one that enables the most accurate estimate of the orientation vector. The choice is described shortly. This orientation estimation is based on a 2BBSS approximation of the volume within that tetrahedron. The eight tetrahedra were chosen because they share the property that three tetrehdron faces are coplanar with faces of the cell which helped simplify the construction of the spline.

For each isosurface mesh vertex, there are two candidate tetrahedra from which the orientation at that vertex could be computed. Next, how our approach decides on the one to use is described. An example situation is shown in Figure 7. In it, the vertex shown in red is located on the rear edge of the cell. One candidate tetrahedron is shown in Figure 7(a) and the other is shown in Figure 7(b). For the case where the vertex lies on an isosurface mesh triangle completely located within a tetrahedron, that tetrahedron is chosen. However, a triangle's surface may span both possible choices of tetrahedra. For such cases, tetrahedron selection is done instead by considering the total number of isosurface mesh triangle edges; we select the tetrahedron containing the greatest number of triangle edges. We have found that selection using this criterion provides more accurate orientation vectors than using a static tetrahedral partition that is ignorant of the triangles's location in the cell. Our approach uses an adaptation of the MC topological case lookup table to record the tetrahedral selections, allowing fast determination of the tetrahedron as well as supporting orientation vector determination coincident with mesh determination (i.e., within an extended MC context).

Next, we describe the orientation determination procedure. The partial derivative of the spline $p(\lambda)$ in the direction ξ_ϕ of a tetrahedron edge $v_\phi - v$ is given by

$$\frac{\partial p}{\partial \xi_\phi} = 2 \sum_{i+j+k+l=1} (a_{i,j+b,k+c,l+d} - a_{i+1,j,k,l}) \lambda_0^i \lambda_1^j \lambda_2^k \lambda_3^l, \quad (7)$$

where $(\lambda_0, \lambda_1, \lambda_2, \lambda_3)$ are the barycentric coordinate variables of the spline equation and (b,c,d) is used to define an offset to a tetrahedral vertex in direction ξ_ϕ.

Figure 8 illustrates the vector calculations when finding the partial derivative in the x direction. The arrows on tetrahedron edges indicate a forward difference calculation using the tetrahedron vertices of that edge. The partial derivative is a linear combination of the differences, with weights for each component dependent on the particular tetrahedral partition being used within the cell. Similar vectors are computed for partial derivatives in the y and z directions.

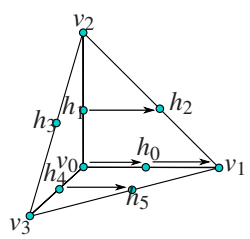


Figure 8: Computing the orientation from sample points

Since cell vertices are located on grid edges, each mesh vertex is guaranteed to have at least two zero-valued components of its barycentric coordinate. By determining which edge type the vertex is located on, we can choose the most appropriate equation to minimize the number of calculations required. For instance, for the tetrahedron shown in Figure 8, the gradient for a vertex on any edge parallel to the base is given by:

$$\nabla F = \left\{ \begin{array}{l} \gamma_0 \sum_{i+j=1} (a_{i,j+1,0,0} - a_{i+1,j,0,0}) \lambda_0^i \lambda_1^j, \\ \gamma_1 \sum_{i+k=1} (a_{i,0,k+1,0} - a_{i+1,0,k,0}) \lambda_0^i \lambda_2^k, \\ \gamma_2 \sum_{i+l=1} (a_{i,0,0,l+1} - a_{i+1,0,0,l}) \lambda_0^i \lambda_3^l \end{array} \right\}, \tag{8}$$

where γ_v, $v = 0,1,2, \gamma = \pm 1$, is an orienting coefficient. For the example in Figure 8, formulation of the spline assumes a tetrahedron oriented as in Figure 8, however the tetrahedral partition used may be a reflection or rotation (or combination of both) of this orientation. The γ coefficient, which corrects for reflected or rotated instances, allows correcting the directions the components of the orientation vector.

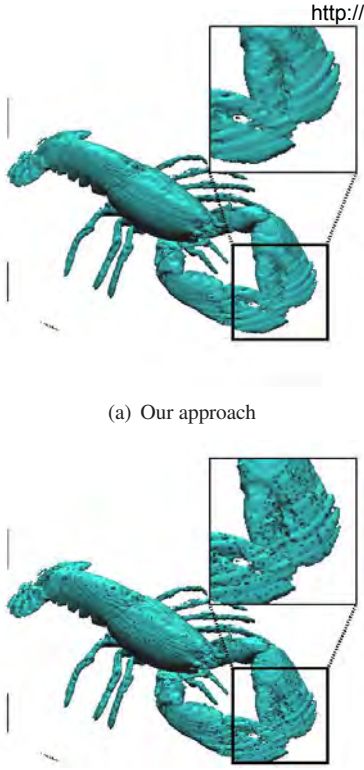
For each tetrahedron the mesh vertex is located in, a gradient vector is produced by evaluating Equation 9. Vertices will be shared among up to four tetrahedra, resulting in as many as four separate vectors per vertex. The orientation vector ultimately assigned to the vertex is the mean of these four gradient vectors.

5 RESULTS

In this section we present results of experiments to evaluate our approach versus LSN. These experiments consider accuracy of orientation vectors and the run times to compute them. We also report a qualitative evaluation of rendered images to determine the impact of degenerate triangles on each approach.

Accuracy was tested by comparing orientation vectors computed using our spline-derived orientations against the orientations using the LSN approach, then comparing these against orientations computed using central-differencing. Eight well-known real (sensed) volume datasets and five mathematically-defined datasets were used in testing. Additionally, we performed visual comparisons of the rendered images to determine if there was a difference between renderings made using the two orientation estimation approaches.

The datasets were converted to sparse grid representations by removing all grid values that were not required by MC to extract the isosurface with marker values. Specifically, grid points that were not on grid edges containing a mesh vertex were set to marker values. By removing all data points that do not contribute to the isosurface extraction, we could operate on volumes with the least possible number of defined values and thus the least favorable datasets for the classic central difference orientation estimation approach used in MC.



(a) Our approach

(b) LSN approach

Figure 9: Renderings performed using both orientation estimation approaches.

5.1 Measurement of Orientation Estimation Accuracy

Isosurfaces were extracted using Marching Cubes for ranges of isovalues on the eight sensed datasets. The range was made large so that results would not be biased against a particular sub-range of isovalues. A root mean square (RMS) error for each isosurface was calculated by comparing the angular difference (in radians) of all orientation vectors produced by both estimation approaches against the central-difference estimate. The central-difference is the baseline in this error comparison because it is equivalent to computing the gradient of a second-order data fitting at each grid point. The mean RMS error of each dataset at all tested isovalues is shown in Table 1. Inspection of individual isovalues on some datasets showed that LSN was sometimes more accurate than our approach, but on average ours appears to be the superior approach. The spline orientation estimation produced more accurate orientation estimates (on average) than the LSN approach in all datasets except for the Engine dataset.

Table 2 shows the RMS errors for 9 isosurfaces extracted on the sensed datasets. LSN does occasionally produce more accurate results, however our orientation

Dataset	Ours	LSN
Foot	0.531	0.547
Frog	0.569	0.589
Lobster	0.369	0.375
MRA	0.639	0.653
Piggy bank	0.876	0.898
Backpack	0.561	0.568
Sheep heart	0.313	0.315
Engine	0.204	0.187

Table 1: Average RMS error of approaches vs. central-difference

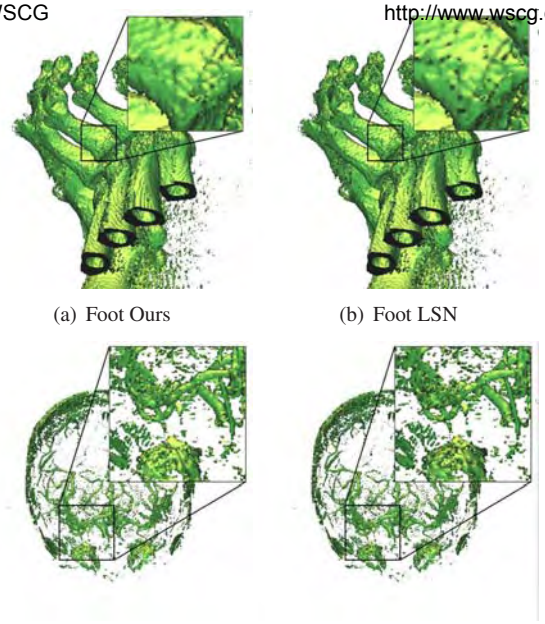
Dataset	Isovalue	Ours	LSN
MRA	65	0.740	0.764
	75	0.684	0.714
	80	0.775	0.783
Foot	80	0.555	0.572
	90	0.502	0.518
	100	0.472	0.322
Frog	40	0.512	0.524
	45	0.513	0.523
	80	0.545	0.640
Lobster	50	0.318	0.316
	65	0.329	0.332
	80	0.336	0.338

Table 2: RMS error of approaches vs. central-difference

estimation produces more accurate results in the majority of cases we tested. Figure 10 shows MRA and Foot isosurfaces (for $\alpha = 65$ and 90, respectively). The magnified callouts show subtle differences in the two renderings, but both are very similar to the baseline images produced using central-difference gradient estimates.

5.2 Accuracy using Mathematically Defined Data

Experiments were also performed to measure the accuracy of the orientation estimation approaches versus exact orientation vector values. These experiments tested scalar fields generated using five mathematically defined fields. The isosurfaces were generated corresponding to level sets (i.e., implicit surfaces) of these fields. Orientation vectors were estimated using our spline-based estimation, the LSN estimation, and the standard MC central-difference approaches. Orientation vectors at each location were compared against the exact orientation vector values computed at the isosurface intersection locations. Table 3 reports the RMS error with respect to the exact orientation vectors for isosurfaces of the zero level set. Excepting the Marschner-Lobb dataset, the central-difference estimates are superior to both LSN and our orientation estimations. But LSN estimates are sometimes better than ours. Thus, empirical evidence suggests that, for mathematically



(a) Foot Ours (b) Foot LSN

(c) MRA Ours (d) MRA LSN

Figure 10: Zoomed comparison of isosurface images

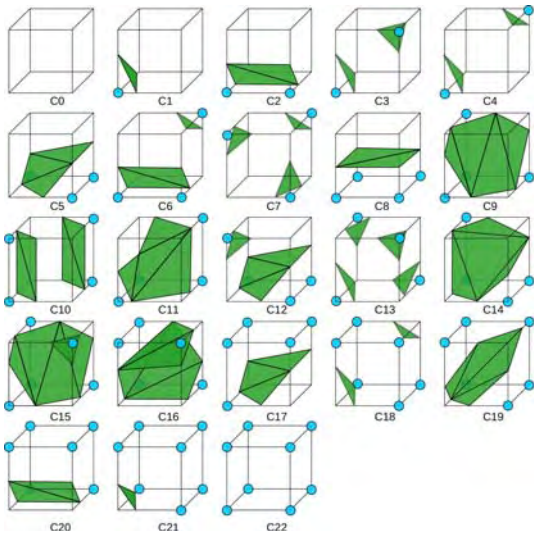


Figure 11: MC lookup table base topologies

defined, noise-free data, LSN estimation may be quite suitable; LSN estimation may provide more accurate normal estimation than our approach for many mathematically defined scalar fields.

5.3 Individual MC Topologies

Since results for the mathematically defined datasets were incongruous with those observed for sensed data (where our approach appears to be better than LSN), we performed an analysis of occurrences of MC base topologies defined in the MC lookup table [12] to determine if one estimation approach produced more accurate orientation vectors for particular base topologies.

Dataset	Ours	LSN	Central Diff
Mar-Lobb	0.0545	0.0616	0.0718
Six Peaks	0.0139	0.0534	0.0179
Genus_3	0.00622	0.00506	0.000265
Flower	0.0261	0.0261	0.0183
Peaks	0.0372	0.0235	0.0218

Table 3: RMS error calculated versus exact orientations

The base topologies are shown in Figure 11 with labels C_i (C_i means "Case i", as used throughout this section). The isosurfaces extracted from mathematically defined datasets showed no occurrences of the Case 4, 7, 12, and 15 topologies. Additionally, very low occurences were observed for Cases 6, 10, 11, 12, 14, 15, 18 and 19. Many of these topologies consist of disconnected triangles within a cell. Due to the nature of the level sets MC produced for these datasets it is not unexpected that occurrences of these topologies would be rare. The sensed data contained far more examples of these topologies. While for some isovalues, there were no instances of a few topological cases, such situations were observed less frequently than for the synthesized datasets. For one dataset (MRA), some isovalues did not give rise to any cells of the type Case 15 or 18. For one dataset (the Engine dataset), the majority of isovalues did not give rise to any of Case 4, 7, 13, or 15 cells. This may be a result of the engine structure in the dataset being manufactured from a CAD model that had a limited number of basic surface types.

To determine the effect that particular base topologies had on orientation estimation accuracy, we considered RMS error of orientation vectors on a topological basis for sensed data isosurfaces. The Case 7, 10 12, 13, 15, and 19 topologies demonstrated much lower RMS errors for our estimation than for LSN estimation. LSN estimation produced consistently more accurate orientation vectors for the Case 8 topology. These results suggest that LSN estimation be considered for isosurfaces likely having low occurrences of topologies beneficial to our approach; mathematically defined datasets similar to ones tested here may be good for LSN.

The LSN's orientation vectors can differ substantially from true orientation vectors and from orientation vectors calculated using central-differencing, as demonstrated in Figure 3. We also analyzed the degree each case should be considered "at-risk" of exhibiting errors due to the incorrect topology assumption, focusing on error-prone vertices. Our criterion for this analysis was if angular divergence in the vector was 90 degrees or more from the central-difference orientation vector. We considered only vertices at the midpoint of cell edges. The analysis showed that 146 of the 256 possible MC cases were potentially problematic. One to five vertices demonstrated angular divergence greater than 90 degrees in these cases. Error comparisons of orientation vectors for just the problematic cases are reported in Table 4 over an average of 100 isovalues for each dataset. Our approach produces orientations that are closer to the central-difference than LSN when these cases are encountered. Figure 9 shows isosurface renderings for the Lobster dataset using both approaches. However, the incidence of orientations that meet the angular divergence criterion in sensed and simulation data is likely much smaller since the triangle vertex locations in the analysis were chosen to highlight the problematic cases and the severity in angular difference is lessened when vertices are located closer to grid point locations.

Dataset	Ours	LSN
MRA	0.639	0.651
Foot	0.922	0.933
Frog	0.652	0.689
Lobster	0.479	0.478

Table 4: RMS error for problematic cases

Dataset	Isoval.	# undef.	Total	%
Foot	40	12204	278894	4.36
Frog	40	1263	101841	1.24
Lobster	40	2946	149250	1.97
Engine	40	4704	637854	0.74
Mar-Lobb	0	0	603343	0
Six Peaks	0	8	2004650	0

Table 5: Undefined orientations using LSN approach

Dataset	Ours (secs)	LSN (secs)
Flower	1.077	2.873
Six Peaks	0.926	2.428
Mar-Lobb	2.959	8.018

Table 6: Orientation estimation times

Rendering artifacts at degenerate triangles in the isosurface mesh can be observed in Figure 9(b). They manifest here as dark spots and are a result of using a vector cross product to compute orientation vectors on degenerate triangles in the virtual mesh. (MC produces a triangle with three coincident vertices when a grid value is identical to the isovalue.) Here, the orientation vector computed for this triangle has length equal to zero. The zero-length vector leads to a zero vector for the Phong illumination diffuse and specular components . Our method does not exhibit this phenomenon, as is illustrated in Figure 9(a), since our orientation vector relies on the result of a fitting to four data values within the cell rather than on any mesh triangles.

In Table 5, we show the number of undefined orientation vectors recorded using the LSN estimation. Volumes with 8 bit integers had more undefined orientations than did those with floating point values. Far fewer undefined orientations were present in the syn-

thetic datasets, which all used 32 bit floating point numbers to store the volume's sample values. The number of undefined orientation vectors using LSN estimation appeared to correspond to the data type used to store the volume's data values.

Finally, in Table 6 execution times for calculating orientations for three of the larger datasets are shown. The LSN estimation requires over twice the computation of our approach. The LSN approach is not as fast as ours.

6 CONCLUSION

We have presented a new approach for estimating isosurface orientation vectors on sparse grid datasets. The typical approach for orientation estimations, central-differencing, cannot be used universally in sparse grids due to undefined data at some grid locations. Our approach can produce isosurface orientations anywhere that MC can produce triangles. Further, the approach is not affected by the presence of degenerate triangles, which produce shading errors in other approaches as a result of undefined orientations. Thus, the new approach has certain advantages even over MC's orientation estimation. Our approach has, on average, a smaller RMS error than a competing approach (using the baseline of central-difference estimations) on real world data. For synthetic data, advantages were less clear. Computation times for our approach were markedly faster. Further, the new approach guarantees orientation vectors to be defined at all vertex locations, making it applicable to a wider variety of data.

An area for further investigation is using spline fittings that observe the continuity properties of super splines in producing more accurate orientation estimations. Also, other isosurfacing algorithms could be investigated with our approach to estimate orientations to determine what increases in accuracy and error tolerance occur. Another area of further investigation is removing random data grid values to simulate random sensor failures. Lastly, we will evaluate the impact of increasing noise levels on the new approach's accuracy.

7 REFERENCES

[1] T. Abdoun, V. Bennett, L. Danisch, T. Shantz, and D. Jang. Field installation details of a wireless shape-acceleration array system for geotechnical applications. In *14th Int'l Symp. on: Smart Structures and Materials & Nondestructive Evaluation and Health Monitoring*. Int'l Society for Optics and Photonics, 2007.

[2] G. Awanou and M.-J. Lai. C^1 quintic spline interpolation over tetrahedral partitions. *Approximation Theory X: Wavelets, Splines, and Apps.*, pages 1–16, 2002.

[3] W. Boehm and A. Müller. On de Casteljau's algorithm. *Computer Aided Geometric Design*, 16:587–605, 1999.

[4] C. D. Correa, R. Hero, and K.-L. Ma. A comparison of gradient estimation methods for volume rendering on unstructured meshes. *IEEE Trans. on Visualization and Computer Graphics*, 17(3):305–319, 2011.

[5] P. de Casteljau. Courbes et surfaces á poles. In *Andres Citröen, Automobiles SA*, Paris, 1963.

[6] S. Djurcilov and A. Pang. Visualizing sparse gridded data sets. *IEEE Computer Graphics and Apps.*, 20:52–57, 2000.

[7] G. Farin. *Curves and Surfaces for CAGD*. Morgan Kaufmann, 5 edition, 2001.

[8] Z. Hossain, U. R. Alim, and T. Möller. Toward high-quality gradient estimation on regular lattices. *IEEE Trans. on Visualization and Computer Graphics*, 17(4):426–439, 2011.

[9] L. Neumann, B. Csébfalvi, A. König, and E. Gröller. Gradient estimation in volume data using 4d linear regression. *Computer Graphics Forum*, 19(3):351–358, 2000.

[10] T. Newman and H. Yi. A survey of the marching cubes algorithm. *Computers & Graphics*, 30(5):854–879, October 2006.

[11] K. Ni, N. Ramanathan, M. N. H. Chehade, L. Balzano, S. Nair, S. Zahedi, E. Kohler, G. Pottie, M. Hansen, and M. Srivastava. Sensor network data fault types. *ACM Trans. on Sensor Networks (TOSN)*, 5(3):25, 2009.

[12] G. M. Nielson, A. Huang, and S. Sylvester. Approximating normals for marching cubes applied to locally supported isosurfaces. In *Proc., Visualization '02*, pages 459–466, 2002.

[13] S. Raman and R. Wenger. Quality isosurface mesh generation using an extended marching cubes lookup table. *Computer Graphics Forum*, 27(3):791–798, May 2008.

[14] C. Rössl, F. Zielfelder, G. Nurnberger, and H.-P. Siedel. Spline approximation of general volumetric data. In *Proc., Symp. on Solid Modeling and Apps.*, pages 71–82, 2004.

[15] K. M. T. Möller, R. Machiraju and R. Yagel. A comparison of normal estimation schemes. In *Proc., Visualization '97*, pages 19–26, 1997.

[16] S. W. Wang and A. E. Kaufman. Volume sampled voxelization of geometric primitives. In *Proc., Visualization '93*, pages 78–84, 1993.

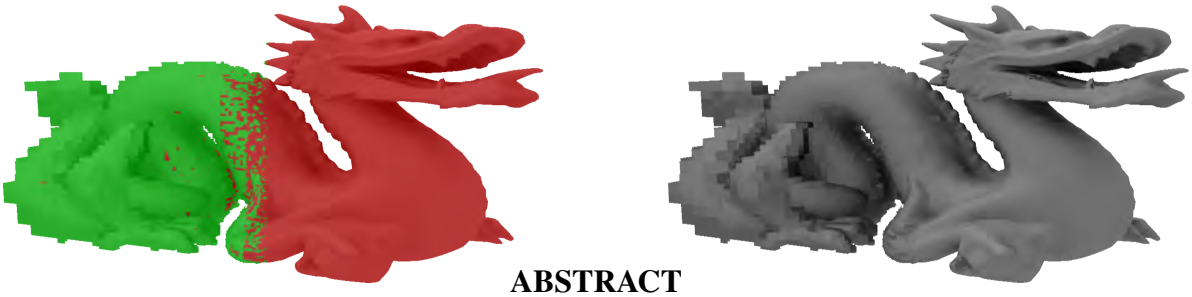
[17] Y. Wang, Y. Liu, and Z. Guo. Three-dimensional ocean sensor networks: a survey. *Journal of Ocean University of China*, 11(4):436–450, 2012.

A Unified Triangle/Voxel Structure for GPUs and its Applications

Martin Weier
Institut of Visual Computing
Sankt Augustin
Grantham-Allee 20
53757, Sankt Augustin, Germany
Martin.Weier@h-brs.de

André Hinkenjann
Institut of Visual Computing
Sankt Augustin
Grantham-Allee 20
53757 Sankt Augustin, Germany
Andre.Hinkenjann@h-brs.de

Philipp Slusallek
Saarland University
Computer Graphics Lab &
Intel Visual Computing
Institute Campus E 1 1
66123 Saarbrücken, Germany
slusallek@cs.uni-saarland.de



ABSTRACT

We present a system that combines voxel and polygonal representations into a single octree acceleration structure that can be used for ray tracing. Voxels are well-suited to create good level-of-detail for high-frequency models where polygonal simplifications usually fail due to the complex structure of the model. However, polygonal descriptions provide the higher visual fidelity. In addition, voxel representations often oversample the geometric domain especially for large triangles, whereas a few polygons can be tested for intersection more quickly.
We show how to combine the advantages of both into a unified acceleration structure allowing for blending between the different representations. A combination of both representations results in an acceleration structure that compares well in performance in construction and traversal to current state-of-the art acceleration structures. The voxelization and octree construction are performed entirely on the GPU. Since a single or two non-isolated triangles do not generate severe aliasing in the geometric domain when they are projected to a single pixel, we can stop constructing the octree early for nodes that contain a maximum of two triangles, further saving construction time and storage. In addition, intersecting two triangles is cheaper than traversing the octree deeper. We present three different use-cases for our acceleration structure, from LoD for complex models to a view-direction based approach in front of a large display wall.

Keywords

Visualization, Computer Graphics, Ray Tracing, Level-of-Detail, Voxelization, Octree, SVO

1 INTRODUCTION

In contrast to polygonal model descriptions, volumetric descriptions are less sensitive to the scene's complexity and enable a progressive refinement – using e.g. octrees, necessary for out-of-core rendering and Level-of-Detail (LoD). However, if these Sparse Voxel Octrees (SVOs) [LK11] are to have a visual quality that compares to a polygonal description, they need a high resolution and require much memory space. When arbitrary scenes are voxelized, many voxels need to be created for single triangles, possibly oversampling the geometric domain even though the polygonal representation is more compact and provides the higher visual fidelity. In addition, it is often cheaper to intersect a couple of triangles compared to traversing an octree deeper. In this paper a hybrid approach is introduced where a SVO is extended with triangle references in the leaf nodes. The voxelization and construction of the structure is entirely performed on the GPU.

Having voxel and polygonal data in one acceleration structure is beneficial because it minimizes manage-

ment and storage cost compared to having two separate structures. In addition, having triangle information in the leaf nodes can reduce the size of the octree. The construction is stopped for those nodes that contain a maximum of two triangles. Two triangles building up a leaf node are often cheaper to intersect than traversing the structure deeper. In addition, they are common for non-isolated triangles, i.e. the ones sharing an edge. Non-isolated triangles form a solid surface and are not crucial to direct geometric aliasing problems. However, the polygonal information provides the higher visual fidelity.

Another benefit of the unified octree structure is that it allows for a convenient smooth intra-level interpolation and color blending between layers in the hierarchy and faster image generation for parts of the scene for which a coarse representation is sufficient.

We contribute by presenting a system to construct and render triangles and voxels in a hybrid acceleration structure. We show how to extend the voxelization method proposed [CG12] and how to perform an interactive construction of unified SVOs on the GPU. In addition, we present a compact data layout allowing for a fast traversal. Finally, we present three applications, where having pre-filtered voxels along with the polygonal information is beneficial and give benchmarks on the construction and traversal times and memory savings by embedding triangle data.

2 RELATED WORK

Several methods have been introduced to create a volumetric description out of a polygonal model, how to construct octrees or multi-level grids and how to traverse these structures.

One important step to generating unified triangle-voxel data is the transformation of the parametric or polygonal description of a model into a volumetric description (voxelization). Early systems such as the Cube system [KS87] try to rebuild a classical hardware-supported rasterization pipeline in software. They use a 3D Bresenham line drawing algorithm to draw the polygonal outline and perform a 3D polygonal filling step. These systems are slow and difficult to implement, as rebuilding an efficient hardware pipeline in software can be challenging.

As dedicated graphics hardware became available to the masses, systems for 3D rasterization using the GPU hardware were proposed. Systems like Voxelpipe [Pan11] and the one proposed by Schwarz and Seidel [SS10] perform voxelization using an optimized triangle/box overlap test on the GPU. The Voxelpipe system allows an A-buffer voxelization where each voxel stores a list of triangles intersecting it. However, using only a triangle/box overlap test creates a binary voxelization of the data, only specifying whether a voxel is on or off. This representation is not sufficient for a LoD representation of textured models. Another example is the system proposed by [ED06] that generates a binary voxelization. However, to use a voxel as a general rendering primitive, more information such as colors and normals are necessary.

Other approaches for performing a surface voxelization on the GPU using a GPU accelerated render pipeline are [DCB+04] and [ZCEP07]. Both approaches render the scene from three sides, combining multiple slices through the model into a final voxel representation. However, rendering a scene multiple times has a negative impact on performance. OpenGL allows to write to a 3D texture or linear video memory directly from the fragment shader. In [CG12], this feature is used to create a boundary voxelization of the model. In this approach, the model has to be rendered only once. Moreover, using the fragment shader means that colors and normals for each voxel are instantly available.

Several methods have been introduced for fast octree and multi-level grid construction. We focus on GPU in-core methods. Each voxel's position in a grid can be represented by a Morton code, that can be used for a fast bottom-up construction of the tree, e.g. in [ZGHG11] [SS10]. A way to create a two level grid is presented in [KBS11]. The algorithm starts by computing pairs of triangle-cell overlaps, sorts these pairs and then fills in the pairs in the grid cells. However, this method must sort the input data first and must be extended to more then two levels.

Another approach is presented by [CG12]. By running multiple shader threads, each voxel is written unsorted top-down to a set of leaf nodes. If a leaf node is touched by a fragment generated in the fragment shader, the node is subdivided further level-by-level. We use a similar approach, and extend it to get an A-Buffer voxelization as well as to construct our hybrid acceleration structure out of it.

A few approaches combine voxel and point based models with polygonal data – one is FarVoxel [GM05]. There, a voxel-based approximation of the scene is generated using a visibility-aware ray-based sampling of the scene represented by a BSP tree. FarVoxels can be used for out-of-core rendering of very large but static models only – the construction of the tree is an offline process. Another approach that combines rasterization and sample-based ray casting is [RCBW12]. In this approach, all the polygonal data is subdivided into cubical bricks, essentially performing a voxelization. However, it is mainly used to speed up rasterization using ray casting methods and not as a general rendering structure.

Sparse Voxel DAGs [KSA13] are an effective way to compact voxel data since they encode identical structures of the SVO in a DAG. However, this method lacks

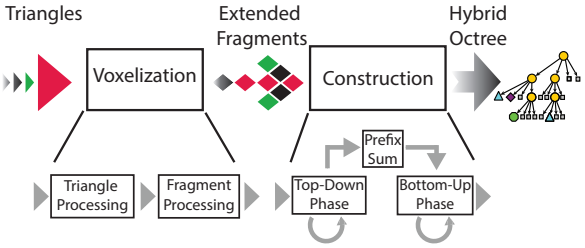


Figure 1: Overview of the GPU-based construction pipeline for the unified structure.

colors and normals for each voxel. If they were to be included, most of the compactness would be gone since colors and normals are unique for most parts of the scene and cannot be easily compacted in a DAG.

3 TRIANGLE/VOXEL STRUCTURE CONSTRUCTION

Our voxelization and octree construction process uses an approach similar to [CG12] using programmable shaders with GLSL. This approach is extended to generate the information on which primitives are touching each non-empty voxel. We show how to use this information to construct the unified acceleration structure on the GPU. Fig. 1 shows the GPU construction pipeline.

MORTON CODE 8B
RGBA 4B
NORMAL 12B
PRIMITIVE ID 4B

Table 1: Structure of an extended fragment entry generated during voxelization, including each element's memory size in byte

Voxelization: The voxelization is performed using OpenGL. The view port's resolution is set to match the voxelized model's target voxel resolution. The view frustum is set up to match the greatest extent of the scene's bounding box. After disabling depth writes and backface culling, each triangle within the view frustum creates a set of fragments accessible in the fragment shader. To extend the projected area of the triangle with respect to the view plane, the triangle is projected to the view plane as if it had been rendered from another side of the bounding box.

Since OpenGL samples each rectangular pixel during the rasterization within the pixel's center, the triangles need to be extended slightly in the geometry shader to ensure that each triangle intersecting a rectangular pixel area covers the pixel's center. This is performed by applying conservative rasterization [HAMO05]. Using the OpenGL Shading Language GLSL and atomic counters, each fragment is written from the fragment shader to a chunk of linear video memory.

Each of these extended fragments stores a position encoded in a Morton code. This enables us to perform a fast per-fragment traversal using bit shifts and a fast comparison of fragments generated at the same spatial position. In addition, the extended fragments store a color, a normal and a triangle index, i.e. the fragment index it originates from. To determine this index, we use the built-in variable `gl_PrimitiveID`. Tab. 1 gives an overview on the memory layout of each extended fragments.

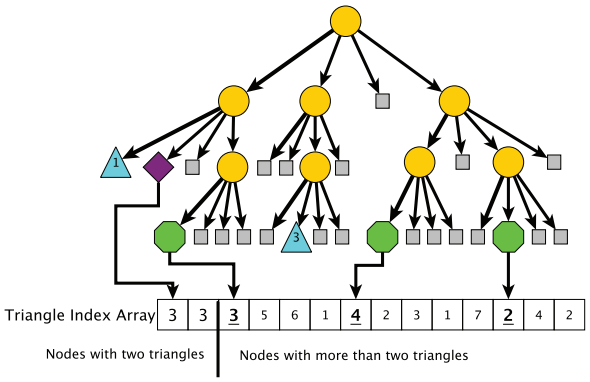


Figure 2: Overview of the unified data structure storing triangles and voxels. Inner nodes (orange), empty nodes (grey), leaf nodes containing a single triangle (light blue), leaf nodes containing two triangles (purple), and leaf nodes containing more than two triangles (green).

Data Structure: Fig. 2 shows the data structure. If a leaf node contains only a single triangle or two triangles, the tree does not need to be constructed for deeper levels for these nodes. If it contains more than two triangles, the node needs to be split. A single node can store the reference to a single triangle alongside with the voxel information. However, if it needs to encode two or more triangles, they are stored in a triangle index array.

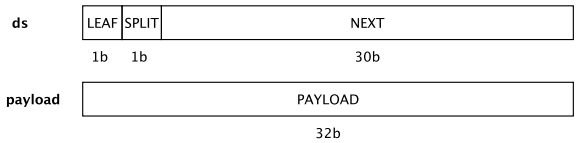


Figure 3: Structure of a single node in the octree.

Each node of the data structure is encoded in two 32 bit fields (see fig. 3). A single bit is used to encode whether the node is a leaf or not, another bit is used to mark a node during construction if it needs to be split further. The next 30 bits either encode the index of the first child node, the id of the triangle if it is the only one represented in the voxel or the index into the triangle index array. The other 32 bits `payload` hold a reference to a voxel array storing the voxel's color, its normal and possibly user-defined fields e.g. material parameters.

Construction: The main idea during construction to decide whether a node contains a single, two or more than two triangles is to cache and compare triangle indices in the 64 bit nodes.

The construction is a splatting process in which several vertex shaders are executed repetitively spanning an arbitrary number of threads using indirect draw calls. First the tree is traversed per-fragment in parallel and construction is done level-by-level. Afterwards the values from the inner nodes of the voxel structure and the color information are written back to the tree nodes bottom-up.

In the first top-down construction phase of the structure, we store the individual triangle IDs from each fragment in the node's two 32 bit fields using atomic comp-and-swap operations. If more than two triangles have to be stored in a node, this node needs to be marked for further splitting. In the next shader step new nodes and voxel payloads for deeper levels are created and the triangle IDs of those nodes that contain only two triangles are written to the triangle index array. Now the first stage is executed again.

Eventually, when the tree is created for the highest resolution, the number of triangles that fell into the leaf nodes are counted using an atomic add operation in the `payload` field. In this stage, each leaf node that has not been already finalized in a earlier shader stage, since it contained only up to two triangles, contains more than two. Afterwards, the triangle counts stored in each leaf node are written to a temporary triangle index count array.

In the next step the prefix sum of the triangle index count array is computed. Finally, the tree is traversed once more and the primitive IDs in the fragment are written to the final array locations in the triangle index array using the triangle index count array and the nodes are relinked accordingly. In this phase we can keep track of the individual primitive id locations in the triangle index array by decrementing the values in the triangle index count array using atomic add operations.

To decide whether a leaf node contains a single, two or more triangles offsets are added to the indices, we store in each leaf node's `next` field. If a node stores an index to a single triangle it encodes the triangle id directly. If it holds an index to a node containing more than two triangles it stores the maximal triangle id plus the index in the triangle index array storing two triangle indices consecutively. If it contains more than two triangles we add the maximal triangle id, the length of the triangle index array storing two triangles and the index. (See fig. 2)

The bottom-up phase continues by filling in the voxel colors, normals and primitive IDs for each node of the tree. Therefore, the tree is traversed per fragment in parallel. Once a shader thread reaches leaf node, the fragment's color and normal must be averaged. This is performed in a similar fashion as in [CG12]. Using an atomic compare-and-swap operation in a loop, each thread checks whether it can write its new summed and averaged value into the voxel's color field. For the normals a simple atomic add on the float components is used. If normals sum up to a zero length normal, e.g. for two opposing faces, the last valid normal is stored.

Finally the tree is processed bottom-up and level by level. Inner nodes are filled by averaging colors and normals and by normalizing the normals of all the child nodes, since the latter resulted only in adding up the normals in the step before.

4 TRIANGLE/VOXEL STRUCTURE TRAVERSAL & INTERSECTION

Rendering of the data structure is performed using a prototypical ray tracer using OpenCL. After the construction, each OpenGL buffer is mapped to OpenCL. These are the buffers containing the nodes, the voxels and the triangle index array and all triangle data, as well as the material information of the model.

Traversal: We decided to implement a traversal using a small stack on the GPU. We set the active parametric t-span of each ray that hits the scene's bounding box to the extent of this bounding box. The algorithm has three phases:

1. If the current first hit voxel within the active t-span is not empty, we traverse the tree deeper and push the parent node with the current t_{max} onto a stack. We set t_{max} to point to the end of the active voxel.
2. If the voxel is empty, we either need to process the next sibling node of the active parent by setting t_{min} to the beginning of the next node within the t-span or,
3. if the node is not a sibling node of the active parent, we need to pop nodes from our stack, reset t_{max} to the position stored on the stack until we can hit the first possible neighboring voxel, and traverse the tree deeper again.

If the traversal reaches a leaf, its triangles can be intersected - either one, two or more. Therefore, the algorithm looks at the index stored in the leaf's `next` field. Since the index is encoded using offsets, it can be decided directly if the node references a single, two or more triangles. The traversal code now determines the closest hit point of the ray and all triangles lying within that leaf node. If the closest triangle is hit and the intersection is within the boundaries described by the leaf node, the traversal returns a structure representing the hit point. Otherwise the traversal is continued with the next sibling node.

Full Octree Resolution						
Scene	Nodes	Triangles	Triangle Index Array	Voxel	**Overall**	
Sponza	42.29	27.66	14.14	46.06	130.15	
Urban Sprawl	18.32	75.19	19.31	20.38	133.21	
Happy Buddha	11.42	103.07	21.94	11.95	148.38	
Forest Scene	30.41	156.25	34.58	33.29	254.53	
Our Method						
Scene	Nodes	Triangles	Triangle Index Array	Voxel	**Overall**	**Saved**
Sponza	10.89	27.66	12.51	13.97	65.03	**50.03%**
Urban Sprawl	12.37	75.19	18.47	14.77	120.81	**9.31%**
Happy Buddha	10.97	103.07	21.92	11.81	147.77	**0.41%**
Forest Scene	21.27	156.25	34.00	27.2	238.72	**6.21%**

Table 2: Size of the acceleration structure (MB). The upper part of the table shows the acceleration structure size of the test scenes for a tree build for all octree levels. The lower part of the table shows our method, where the tree is built only for nodes containing more than two triangles.

Inter-level blending: For the LoD selection and to enable a smoother blending between different levels of the hierarchy we use Ray Differentials [Ige99]. Each ray is represented by its origin and a unit vector describing its direction. In addition, we store its' differentials describing the pixel offset on the image plane in x and y direction.

By using ray differentials, we can compute an estimated pixel's footprint in world space on the voxels. This footprint can be compared with the size of an individual voxel at level l. If the pixel's footprint is roughly equal or smaller than the voxel, we can stop traversing deeper.

In addition, we compute a value describing the underestimation $i(l,f)$ of the size of the pixel's footprint and the actual size of nodes at level l and $l-1$ by computing

$$i(l,f) = \frac{2 \cdot v_w(l) - f}{v_w(l)}$$

with $v_w(l)$ being the length of a side of a voxel in world space and f being the estimated length of the pixel's footprint at the ray's hit point. This value can be used as interpolation factor between the two subsequent levels in the SVO. Since we traverse the tree using a small stack, we can keep track of the voxel at level $l-1$ directly and use the interpolation factor during shading and lighting computations.

5 BENCHMARKS

The benchmarks of our system were performed using a Nvidia GeForce GTX Titan with 6GB video memory on an Intel Core i7 system with 16GB RAM. Fig. 4 shows the construction times of four different test scenes. The forest test scene shows 13 highly detailed plant models on a small plane. As expected, increasing triangle counts increase the run time of the construction. However, the pure triangle count is not the only parameter when it comes to measuring construction times as highly detailed textures and shaders extend the time it takes to voxelize the model.

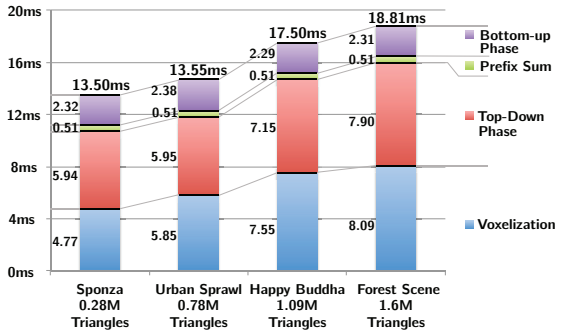


Figure 4: Run times for each phase of the construction as well as the overall construction time. Each scene was voxelized with a resolution of 512^3.

Scene	Voxel only	Triangle only	**Hybrid Structure**
Sponza	57.3 fps	18.2 fps	20.6 fps
Urban Sprawl	40.3 fps	13.3 fps	23.7 fps
Happy Buddha	63.1 fps	10.1 fps	16.7 fps
Forest Scene	64.2 fps	2.4 fps	12.9 fps

Table 3: Avg. fps of four different scenes rendered with a resolution of 1024×1024 using only primary rays and phong lighting with simple shadows and a single point light source. Each scene was voxelized with a resolution of 512^3.

Table 2 shows the advantage of our method in comparison to a full build of the octree without stopping the construction early in terms of size of the acceleration structure. Both versions store the triangles in their leaf nodes as a reference to the `triangle index array`. We have included the size needed to store the triangles themselves, which largely depends on the scene. The triangle count in the Sponza scene is very low. If one only considers the size of the nodes and the voxel data, the overall saved space amounts to a larger percentage for most scenes. The Happy Buddha scene has many, but very small triangles. For this scene construction can't be stopped for most inner nodes resulting in only a small memory saving.

We have rendered all scenes with a resolution of 1024×1024 using a typical fly through for about 700 frames and averaged the run times. The results in tab. 3 show the rendering times from the OpenCL renderer shooting primary rays with phong lighting, a single point light source and no texture filtering. Rendering only voxels is fast but lacking visual quality. Traversing our structure displaying triangles only provides the highest visual quality but is slow an offers no LoD - aliasing can occur. he hybrid structure provides a good trade-off in speed and offers LoD.

However, measuring the frame rates for the hybrid approach is non trivial since they increase drastically if parts of the scene show the voxel data only. For scenes like Sponza showing an atrium where a camera is mostly "inside" the model, only a few camera positions can make use of the voxel data, resulting in only a small speed up. In the Forest- or the Urban Sprawl scene parts of the model are in the distance more often. Thus the voxel data is used more frequently resulting in larger speed ups.

6 APPLICATIONS

Our hybrid structure is well-suited for applications that need a general LoD scheme, since the regular voxel description allows to create a representation for arbitrary input meshes. In principle the hybrid structure can be seen as a multi level grid, omitting the fact that this structure contains a color and a normal for each grid cell. However, this additional information, is well suited to some scenarios to reduce aliasing and speed up rendering. We present three different applications: a visualization of large outdoor scenes, urban environments and a view-direction based rendering approach in front of a large tiled display wall.

The first application (cf. fig. 6a) uses the hybrid acceleration structure to render highly complex vegetated areas with LoD. Here far distant models project to only a few pixels on screen creating aliasing artifacts. We use an approach similar to [DMS06] [WHDS13]. On the highest level a nested hierarchy of kd-trees over wang tiles with Poisson Disc Distributions is used to represent plant locations resulting in instanced, but aperiodic repetitions. Each scene contains millions of highly complex plant models reused throughout the scene.

The advantage of our hybrid representation over a polygonal simplification is that, within a regular octree structure, an approximation of high-frequency input models such as trees with different LoDs can be generated. Polygonal simplification of such models usually fails due to the complex foliage and branching structure of the trees. Sample caching strategies in object space that provide LoD are limited to single instances, e.g. samples can't be cached in the accelerations structure of a single tree since it is reused. Therefore, it is beneficial to have pre-filtered voxel data at hand to limit aliasing artifacts or to reduce the oversampling needed to create smooth animations and crisp images. In addition, this speeds up rendering. We can render a scene with trillions instantiated triangles consisting of 40Mio. trees at a resolution of 720p with about 5-7fps including direct shadows using our prototypical OpenCL ray caster.

Another example where this LoD structure is beneficial are urban scenes as shown in fig. 5a and fig. 5b. Even though a polygonal simplification of such structures is not as challenging as for tree models, renderings of such scenes from far away have to cope with high-frequency aliasing. If this urban scene is viewed from a distance, the highly varying z-depth of the scene generate geometric aliasing which can be reduced by having a pre-filtered voxel structure. Moreover, voxel are independent from the scenes local complexity. In addition, possibly large triangles in such a scene further reduce the size of the octree. Furthermore, the hybrid structure allows for smoother transitions and color blending between different layers of the hierarchy (cf. fig. 5b) and faster render times for highly detailed parts in the scene that are viewed from the distance.

A further application is shown in fig. 6b. There the structure is used for a view dependent rendering on a large tiled display wall. Since coarse voxel representations can be renderer faster than highly complex polygonal models, the voxel representation is mainly used to speed up rendering.

The user's central field of view is tracked and rendered in high quality using the polygonal representation, whereas the surrounding is rendered using our LoD approach. Therefore, we compute an intersection of the tracked user's view frustum with a virtual display wall. Using the intersections an ellipsoid is generated. Points within this ellipsoid are rendered with maximal resolution using polygonal data. For points outside of the ellipsoid the distance from the ellipsoid to the current pixel is computed. This distance is use to decide whether a deeper traversal of the hierarchy is necessary or if traversal can be stopped early. The transitions between the layers of the hierarchy are blurred using a post-processing step in image space.

7 DISCUSSION

We presented an approach to building a hybrid acceleration structure storing voxels for inner nodes, stopping construction of deeper levels if the number of primitives within that node are not larger than two and storing the full triangle list for each leaf node that represents the finest voxelized level. This way, we generate a LoD description of the input geometry. The advantage of this representation over a polygonal simplification is that, within a regular octree structure, we generate a good

(a) (b)

Figure 5: Rendering of an urban environment (5a) using our unified octree structure with voxel data in the background. Fig. (5b) shows a color coding. The red areas were rendered using polygonal data and the green regions were rendered using voxels.

(a) (b)

Figure 6: Rendering of instantiated tree models (6a) and a focus and context based rendering in front of a large display wall (6b) using our unified acceleration structure.

approximation of high-frequency input models such as trees. In addition, this speeds up rendering by providing a coarse representation for areas that are of minor interest in a visualization or are not visible/noticeable to the user. Since the construction on the GPU is performed in-core, the resolution of the voxelization is limited. However, the system is fast enough to construct an octree of a scene in real time doing a complete rebuild.

One problem targeted by further research is that an octree is not truly adaptive with respect to the scene's input geometry. If one has highly complex geometry inside a single leaf voxel, traversing these parts of the scene can have a huge impact on performance. Simply building a tree deeper by a regular subdivision of these parts, is often not sufficient to divide the model's input geometry. It would be better to either identify these high resolution parts beforehand and voxelize them separately or automatically use truly adaptive acceleration structures such as BVHs or kD-Trees for these parts of the scene. However, since a coarser voxel representation is available, the renderer can decide to stop traversing these parts and display the coarse voxel representation to stay within a constant frame rate. In addition, due to the regularity of the octree's structure, more advanced optimizations such as e.g., a beam optimization [LK11] could be applied. Moreover, for improved GPU utilization, it might be beneficial to postpone the triangle intersection from inside the octree traversal to subsequent rendering passes.

Another aspect crucial to performance is memory management. Since the number of fragments generated by the voxelizer, the size of the octree and the triangle index list are not known in advance, buffers must either be preallocated with a maximal size, be used in a caching scheme (e.g. [CNLE09]), or more advanced memory management must be applied – though determining the size needed for buffers, is a problem most grid construction algorithms have in common. However, once we have generated the voxel's extended fragment list, our approach can stop the octree construction early when too much memory is needed to construct deeper levels. The system has been extended to perform an out-of-

core voxelization and construction for parts of the scene that have to be voxelized with a higher resolution.

Voxel structures have disadvantages which should be targeted by further research. It is merely possible to average different material informations inside a singe voxel cell. Furthermore, due to their grid like structure, shadows are hard to implement because neighboring voxels tend to cast shadows on themselves. These shadows create a high-frequency noise in the image which is disadvantageous if one wants to use voxels to reduce aliasing. Another issue is the size of the structure. However, we have shown that our structure is compact enough to represent several dozens of models, voxelized with a high-resolution, in GPU memory at once.

8 REFERENCES

[CG12] Cyril Crassin and Simon Green. *Octree-Based Sparse Voxelization Using The GPU Hardware Rasterizer*. OpenGL Insights. NVIDIA Research, July 2012.

[CNLE09] Cyril Crassin, Fabrice Neyret, Sylvain Lefebvre, and Elmar Eisemann. Gigavoxels : Ray-guided streaming for efficient and detailed voxel rendering. In *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D)*, Boston, MA, Etats-Unis, feb 2009. ACM, ACM Press. to appear.

[DCB+04] Zhao Dong, Wei Chen, Hujun Bao, Hongxin Zhang, and Qunsheng Peng. Real-time voxelization for complex polygonal models. In *Proceedings of the Computer Graphics and Applications, 12th Pacific Conference*, PG '04, pages 43–50, Washington, DC, USA, 2004. IEEE Computer Society.

[DMS06] Andreas Dietrich, Gerd Marmitt, and Philipp Slusallek. Terrain guided multilevel instancing of highly complex plant populations. In *Proceedings of the 2006 IEEE Symposium on Interactive Ray Tracing*, pages 169–176, September 2006.

[ED06] Elmar Eisemann and Xavier Décoret. Fast scene voxelization and applications. In *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pages 71–78. ACM SIGGRAPH, 2006.

[GM05] Enrico Gobbetti and Fabio Marton. Far voxels: A multiresolution framework for interactive rendering of huge complex 3d models on commodity graphics platforms. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, pages 878–885, New York, NY, USA, 2005. ACM.

[HAMO05] Jon Hasselgreen, Tomas Akenine-Möller, and Lennart Ohlsson. *GPU Gems 2: Conservative Rasterization*, volume 2, chapter 42, pages 677–690. NVIDIA, Addison-Wesley, 2005.

[Ige99] Homan Igehy. Tracing ray differentials. pages 179–186, 1999.

[KBS11] Javor Kalojanov, Markus Billeter, and Philipp Slusallek. Two-level grids for ray tracing on gpus. *Comput. Graph. Forum*, 30(2):307–314, 2011.

[KS87] Arie Kaufman and Eyal Shimony. 3d scan-conversion algorithms for voxel-based graphics. In *I3D '86 Proceedings of the 1986 workshop on Interactive 3D graphics*, pages Pages 45 – 75. ACM New York, NY, US, 1987.

[KSA13] Viktor Kämpe, Erik Sintorn, and Ulf Assarsson. High resolution sparse voxel dags. *ACM Trans. Graph.*, 32(4):101:1–101:13, July 2013.

[LK11] Samuli Laine and Tero Karras. Efficient sparse voxel octrees. *IEEE Transactions on Visualization and Computer Graphics*, 17:1048–1059, 2011.

[Pan11] Jacopo Pantaleoni. Voxelpipe: A programmable pipeline for 3d voxelization. In *Proceedings of the ACM SIGGRAPH Symposium on High Performance Graphics*, HPG '11, pages 99–106, New York, NY, USA, 2011. ACM.

[RCBW12] Florian Reichl, Matthäus G. Chajdas, Kai Bürger, and Rüdiger Westermann. Hybrid sample-based surface rendering. In *Proceedings of VMV 2012*, pages 47–54, 2012.

[SS10] Michael Schwarz and Hans-Peter Seidel. Fast parallel surface and solid voxelization on gpus. *ACM Trans. Graph.*, 29(6):179:1–179:10, December 2010.

[WHDS13] Martin Weier, André Hinkenjann, Georg Demme, and Philipp Slusallek. Generating and rendering large scale tiled plant populations. *JVRB - Journal of Virtual Reality and Broadcasting*, 10(1), 2013.

[ZCEP07] Long Zhang, Wei Chen, David S. Ebert, and Qunsheng Peng. Conservative voxelization. *Vis. Comput.*, 23(9):783–792, August 2007.

[ZGHG11] Kun Zhou, Minmin Gong, Xin Huang, and Baining Guo. Data-parallel octrees for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 17(5):669–681, May 2011.

Magnetic Resonance Images Reconstruction using Uniform Discrete Curvelet Transform Sparse Prior based Compressed Sensing

Bingxin Yang Min Yuan Yide Ma* Jiuwen Zhang
LanZhou University
Tianshui South Road No.222, 730000, LanZhou, China
* Corresponding author
18919849359@163.com yuanm@lzu.edu.cn ydma01@126.com zhangjw@lzu.edu.cn

ABSTRACT

Compressed sensing(CS) has shown great potential in speeding up magnetic resonance imaging(MRI) without degrading images quality. In CS MRI, sparsity (compressibility) is a crucial premise to reconstruct high-quality images from non-uniformly undersampled *k*-space measurements. In this paper, a novel multi-scale geometric analysis method (uniform discrete curvelet transform) is introduced as sparse prior to sparsify magnetic resonance images. The generated CS MRI reconstruction formulation is solved via variable splitting and alternating direction method of multipliers, involving revising sparse coefficients via optimizing penalty term and measurements via constraining *k*-space data fidelity term. The reconstructed result is the weighted average of the two terms. Simulated results on in vivo data are evaluated by objective indices and visual perception, which indicate that the proposed method outperforms earlier methods and can obtain lower reconstruction error.

Keywords

Compressed sensing, magnetic resonance imaging, uniform discrete curvelet transform, variable splitting, alternating direction method of multipliers.

1 INTRODUCTION

Traditional scanning methods of magnetic resonance imaging(MRI) spent plenty of time on data acquisition. This brought negative influences for clinical diagnosis. *K*-space undersampling provides one method to speed up the imaging at the expense of introducing aliasing for violating the Nyquist (Shannon) sampling theorem.

Compressed sensing(CS) [baraniuk2007compressive, 1614066] points out, sparse or compressible signal can be reconstructed precisely from less number of sampled data than those constrained by Nyquist sampling theorem. Hence, CS provides theoretical feasibility for highly undersampled MR images reconstruction. The emerging approach is termed CS MRI [lustig2007sparse, 4472246]. The main principles of CS MRI are that the images to be reconstructed can be sparsely represented; measurement matrix is irrelevant to sparse transform basis; the reconstruction optimization problem can be solved by using nonlinear method. In CS MRI, incoherent random, radial and spiral sampling trajectories are applied to obtain *k*-space measurements [lustig2007sparse, chen2010novel, santos2006single]. The generally employed sparsifying methods include spatial finite-difference [lustig2007sparse, huang2011efficient, huang2012compressed], discrete wavelet transform(DWT) [lustig2007sparse, huang2011efficient, huang2012compressed], multi-scale geometric analysis(MGA) methods (contourlet transform [1532309], nonsubsampled contourlet transform [da2006nonsubsampled], sharp frequency localization contourlet(SFLCT) [lu2006new], discrete curvelet transform using fast algorithm(FDCT) [candes2006fast] and discrete shearlet transform(DST) [lim2010discrete]), dictionary learnt from intermediate reconstruction or fully sampled images [ning2013magnetic, qu2012undersampled], temporal sparsity along temporal axis for dynamic cardiac imaging [bilen2012high] and the combination of some of these transforms [lustig2007sparse, huang2011efficient]. The main thoughts of reconstruction approaches are nonlinearly reconstructing original signal accurately from a small number of measurements. The generally used are greedy pursuit class (matching pursuit, orthogonal

matching pursuit) for solving sparse coefficients l_0 regularization, provided that the sparsity of image is already known; linear programming (gradient projection, basis pursuit) handling sparse coefficients l_1 regularization at the cost of high computational complexity; minimizing non-convex l_p $(0 < p < 1)$ quasi-norm such as the recent one in [candes2008enhancing], which doesn't always give global minima and is also slow. The widely used methods are based on augmented Lagrangian for solving convex, non-smooth regularization (total variation and l_1) optimization. These methods include YALL1 [yang2011alternating], FCSA [huang2011efficient], split augmented Lagrangian shrinkage algorithm(SALSA) [afonso2010fast] and constrained split augmented Lagrangian shrinkage algorithm(C-SALSA) [5570998], etc.

In this paper, a novel MGA method termed uniform discrete curvelet transform(UDCT) (refer to [5443489] for details) is adopted to sparsify MR images. In terms of the alias free subsampling in frequency domain they both employed, UDCT has similar properties as wrapping-based FDCT, such as tight frame property, highly directional sensitivity and anisotropy. Besides, UDCT is superior than FDCT for its lower redundancy of 4 and clear coefficients parent-children relationship. Reconstruction model is proposed involving UDCT coefficients regularization term and k-space data fidelity term. To solve the corresponding reconstruction model, C-SALSA, i.e., variable splitting(VS) and alternating direction method of multipliers(ADMM-2) [5570998] is used. The proposed CS MRI method is termed UDCSMRI.

The paper is organized as follows. Section 2 describes the existing CS MRI work, and then introduces UDCSMRI in detail including UDCT sparse prior and corresponding reconstruction model handling the ill-posed linear inverse problems. In section 3 UDCSMRI is compared with current CS MRI methods in reconstruction performance. Then its ability of handling noise and convergence performance is analyzed. Conclusions and future work involving extending this work to dynamic parallel MRI are explicit in section 4.

2 MATERIALS AND METHODS

CS MRI

Define $\mathbf{x} \in \mathbb{C}^{\mathbf{n}}$ is vector-version of 2D image to be reconstructed. $\mathbf{y} = \mathbf{F_u x}$ denotes undersampling in k-space, where $\mathbf{F_u} \in \mathbb{C}^{\mathbf{m}\times\mathbf{n}}$ means undersampled Fourier Encoding matrix and $\mathbf{y} \in \mathbb{C}^{\mathbf{m}}$ represents k-space measurements. $\mathbf{\Psi} \in \mathbb{C}^{t\times n}$ represents analytical sparse transform matrix or the inverse of a set of learnt signals. CS reconstructs the underlying MR image $\mathbf{x}$ from measurements $\mathbf{y}$ via solving the constrained linear inverse problem, denoted as Eq. (1)

$$\min_{\mathbf{x}} \|\mathbf{\Psi x}\|_1 \text{ s.t. } \|\mathbf{F_u x} - \mathbf{y}\|_2^2 \leq \boldsymbol{\varepsilon} \tag{1}$$

where $\boldsymbol{\varepsilon} \in \mathbb{C}^m$ controls the allowed noise level in reconstructed image, l_1 enforces sparsity, l_2 constrains the data fidelity. Finite-difference (total variation) is generally added to the objective to suppress the noise and preserve images details simultaneously, then the problem is

$$\min_{\mathbf{x}} \|\mathbf{\Psi x}\|_1 + \beta TV(\mathbf{x}) \text{ s.t. } \|\mathbf{F_u x} - \mathbf{y}\|_2^2 \leq \boldsymbol{\varepsilon} \tag{2}$$

where $\beta > 0$ denotes weight of total variation(TV). Rather than Eq. (1), most current methods handling linear inverse problems with convex, non-smooth regularization (l_1 and TV) consider the unconstrained problem

$$\min_{\mathbf{x}} \beta_1 \|\mathbf{\Psi x}\|_1 + \beta_2 TV(\mathbf{x}) + \frac{1}{2}\|\mathbf{F_u x} - \mathbf{y}\|_2^2 \tag{3}$$

in which $\beta_{1(2)} > 0$ is regularization parameter. The commonly used techniques dealing with Eq. (3) are VS and methods upon augmented Lagrangian, such as TVCMRI [ma2008efficient], RecPF, FCSA, SALSA, etc. However, Eq. (3) is not efficient for ignoring $\boldsymbol{\varepsilon}$, which has a clear meaning (proportional to the noise deviation) and is easier to set than parameter $\beta_{1(2)}$. Additionally, numerous different reconstruction models have been explored, such as NLTV-MRI incorporating with nonlocal TV [huang2012compressed], reconstruction upon wavelet tree structured sparsity(WaTMRI) studied in [NIPS20124630], reconstruction by using dictionary learning(DL) [qu2012undersampled, ning2013magnetic] and patch-based nonlocal operator combined with VS and quadratic penalty reconstruction technique named PANO [qu2014magnetic], etc. Besides, 3D dynamic parallel imaging has also been proposed and is of great significance for practical MRI applications. It is established on either sparsity along temporal axis [bilen2012high] or structured low-rank matrix completion [shin2013calibrationless],.

Proposed Method based on UDCT

In this paper, MR images are sparsified by MGA method named UDCT. Efficient C-SALSA is introduced to solve the generated CS MRI reconstruction formulation under UDCT sparse prior. MR image $\mathbf{x}$ to be reconstructed is initialized to one zero-filling image. This zero-filling image is obtained from the result of direct inverse Fourier transform to zero filled k-space measurements, represented as $\mathbf{x_0} = \mathbf{F_u^H y}$. Zero-filling image serves as the original intermediate image. The real and imaginary part of $\mathbf{x_0}$ are decomposed into J levels by using UDCT separately, $2\kappa_j$ directional sub-bands for each level. CS MRI reconstruction problem comes down to solving the optimization problem constrained by image transform sparsity and k-space measurements fidelity (in an iterative process). The solving process requires the definition of the Moreau proximal maps of regularization term and fidelity term.

Reconstruction result is the trade-off between the two terms and then serves as the intermediate image for the next iteration. This procedure is executed iteratively until some stop criterion is satisfied. Framework of UDCSMRI in Fig.1 demonstrates clearly the implementation process.

Uniform Discrete Curvelet Transform

As is known, discrete wavelet basis only represents the location and features of singular point with limited directions. The generally used contourlet transform lacks shift-invariance and brings pseudo-Gibbs phenomena around singular points. NSCT owns too high redundancy and SFLCT cannot capture clear directional features in spite of flexible redundancy. The needle-shaped elements of FDCT allow very high directional sensitivity and anisotropy and are thus very efficient in representing line-like edges. But FDCT possesses too high redundancy, which makes it sub-optimal in sparse representation, either. UDCT has been proposed as an innovative implementation of discrete curvelet transform for real-valued signals. Utilizing the ideas of FFT-based discrete curvelet transform and filter-bank based contourlet transform, UDCT is designed as a perfect multi-resolution reconstruction filter bank(FB) but executed by FFT algorithm. The number of UDCT coefficients are fixed at each scale and sizes of directional sub-bands are the same for each scale, which provides simple calculation. UDCT can provide a flexible instead of fixed number of clear directions at each scale to capture various directional geometrical structures accurately. Besides, the forward and inverse transform form a tight and self-dual frame with an acceptable redundancy of 4 to allow the input real-valued signal to be perfectly reconstructed. UDCT has asymptotic approximation properties: for image $\mathbf{x}$ with C^2 (C is a constant) singularities, the best N-term approximation $\mathbf{x_N}$ (N is the number of most important transform coefficients allowing reconstruction) in the curvelet expansion is [candes2000curvelets]

$$\|\mathbf{x}-\mathbf{x_N}\|_2^2 \leq CN^{-2}\left(\log N\right)^3 \ N \to \infty \tag{4}$$

This property is known as the optimal sparsity. Therefore, UDCT is considered as the preeminent MGA method for CS MRI application.

Constrained Split Augmented Lagrangian Shrinkage Algorithm

Define Φ as regularization function, $\mathbf{\Psi}$ the UDCT analytical operator, the sparse representation is defined as $\boldsymbol{\alpha} = \mathbf{\Psi x}$. The reconstruction model can thus be denoted as

$$\min_{\boldsymbol{\alpha},\mathbf{x}} \Phi\left(\boldsymbol{\alpha}\right) = \begin{cases} \|\boldsymbol{\alpha}\|_1 & \text{if } \Phi = l_1 \\ TV\left(\mathbf{\Psi}^{-1}\boldsymbol{\alpha}\right) & \text{if } \Phi = TV \end{cases} \tag{5}$$
$$\text{s.t. } \|\mathbf{F_u x}-\mathbf{y}\|_2^2 \leq \boldsymbol{\varepsilon}$$

Eq. (5) is solved by C-SALSA. Different from the previous augmented Lagrangian based methods to solve Eq. (3), C-SALSA has been proposed as a new augmented Lagrangian based method, which directly solves the original constrained inverse problem optimization efficiently. C-SALSA first translates the constrained Eq. (5) into an unconstrained one via adding the indicator function of the feasible set, the ellipsoid $\{\mathbf{x} : \|\mathbf{F_u x}-\mathbf{y}\|_2^2 \leq \boldsymbol{\varepsilon}\}$, to the objective in Eq. (5). Then the unconstrained problem can be denoted as

$$\min_{\boldsymbol{\alpha},\mathbf{x}} \lambda_1 \Phi\left(\boldsymbol{\alpha}\right) + \lambda_2 \mathscr{L}_{E(\boldsymbol{\varepsilon},\mathbf{I},\mathbf{y})}\left(\mathbf{F_u x}\right) \tag{6}$$

In Eq. (6), parameters λ_1 and λ_2 measure the weight of the regularization term and error constraint term, respectively. The values linearly increase along with the increase of iteration number ($\lambda_{1(2)} \leftarrow \rho\lambda_{1(2)}$, $\rho > 1$ means linear growth factor). Eq. (6) is translated into another constrained problem via VS, denoted as

$$\min_{\boldsymbol{\alpha}\in\mathbb{C}^t,\mathbf{x}\in\mathbb{C}^{\mathbf{n}},\boldsymbol{\nu}\in\mathbb{C}^{\mathbf{m}}} \lambda_1 \Phi\left(\boldsymbol{\alpha}\right) + \lambda_2 \mathscr{L}_{E(\boldsymbol{\varepsilon},\mathbf{I},\mathbf{y})}\left(\boldsymbol{\nu}\right) \text{ s.t. } \boldsymbol{\nu} = \mathbf{F_u x} \tag{7}$$

Finally, ADMM-2 solves the two sub-problems concerning $\boldsymbol{\alpha}$ and $\boldsymbol{\nu}$. The reconstruction result is obtained in this way. In terms of sub-problem concerning the regularization Φ, the Moreau proximal mapping function can be defined as

$$\mathbf{\Theta}_{\Phi}\left(\widehat{\boldsymbol{\alpha}}\right) = \arg\min_{\boldsymbol{\alpha}} \frac{1}{2}\left\|\boldsymbol{\alpha}-\widehat{\boldsymbol{\alpha}}\right\|_2^2 + \Phi\left(\boldsymbol{\alpha}\right) \tag{8}$$

where $\widehat{\boldsymbol{\alpha}}$ is the result of mapping to $\boldsymbol{\alpha}$ according to the mapping relation $\mathbb{C}^t \to \mathbb{C}^t$. If $\Phi\left(\cdot\right) \equiv \|\cdot\|_1$, $\mathbf{\Theta}_{\Phi}$ is simply a soft threshold. If Φ is TV norm, Chambolle's algorithm [chambolle2004algorithm] is available to compute the involving problem. $E(\boldsymbol{\varepsilon},\mathbf{I},\mathbf{y})$ represents a closed $\boldsymbol{\varepsilon}$-radius Euclidean ball centered at $\mathbf{y}$. The Moreau proximal map of $\mathscr{L}_{E(\boldsymbol{\varepsilon},\mathbf{I},\mathbf{y})}$ can be simply denoted as the orthogonal projection of $\boldsymbol{\nu}$ on the closed $\boldsymbol{\varepsilon}$-radius ball centered at $\mathbf{y}$

$$\mathbf{\Theta}_{\mathscr{L}_{E(\boldsymbol{\varepsilon},\mathbf{I},\mathbf{y})}}\left(\boldsymbol{\nu}\right) = \begin{cases} \mathbf{y} + \boldsymbol{\varepsilon}\frac{\boldsymbol{\nu}-\mathbf{y}}{\|\boldsymbol{\nu}-\mathbf{y}\|_2} & \text{if } \|\boldsymbol{\nu}-\mathbf{y}\|_2^2 > \boldsymbol{\varepsilon} \\ \boldsymbol{\nu} & \text{if } \|\boldsymbol{\nu}-\mathbf{y}\|_2^2 \leq \boldsymbol{\varepsilon} \end{cases} \tag{9}$$

The resulting algorithm is summarized in Algorithm C-SALSA-2 [5570998].

3 EXPERIMENTAL RESULTS AND ANALYSIS

Experimental Setup

The reconstruction performance of UDCSMRI for various MR raw data, is analyzed from four aspects. Experimental raw data include complex-valued T2-weighted brain image (MR T2wBrain_slice27 of 256×256), water phantom [ning2013magnetic], real-valued MBA_T2_slice006, randomly selected

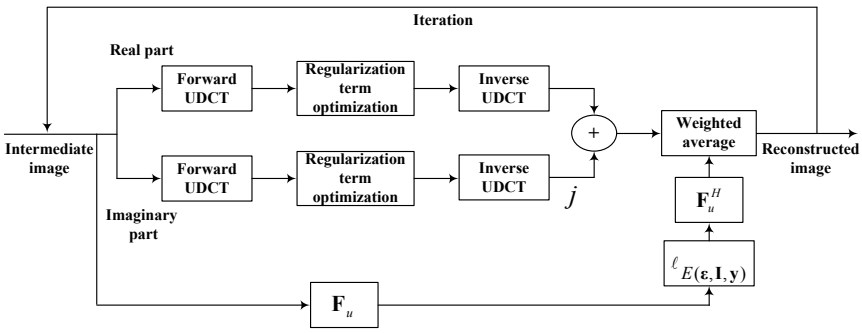


Figure 1. Framework of UDCT based CS MRI

AIDS dementia (slice 0-16), Brain Tumor (slice 0-23) and Normal aging (slice 0-53) (Courtesy of http://www.med.harvard.edu/AANLIB/home.html). Partial raw images and sampling schemes are shown in Fig.2. Computations are performed on a 64-bit Windows 7 operating system with an Intel Xeon E5 CPU at 2.80 GHz and 8 GB memory, MATLAB R2011b. Numerical metrics of quality assessment for reconstructed images are peak signal-to-noise ratio(PSNR) (in dB) and relative l_2 norm error(RLNE) [qu2012undersampled].

Comparison with Earlier Methods

The performance of UDCSMRI for images in Fig.2(a)-(c) is compared with that of TVCMRI, FCSA and WaTMRI. UDCT decomposition of $J = 1$, 12 directional sub-bands for each scale is adopted by Fig.2(a)-(b). For Fig.2(c), UDCT decomposition of $J = 3$, 12 directional sub-bands for each scale is used. The preset maximum iteration number for ADMM-2 is $K = 70$.

MR T2wBrain_slice27 reconstruction under 40% Cartesian sampling scheme is exhibited in Fig.3. Fig.3 indicates that reconstructed images under wavelet basis sparse regularization show severe pseudo-Gibbs phenomena, edge blur and aliasing. Whereas UDCSMRI with $\Phi = l_1$ (UDCSMRI(l_1)), UDCSMRI with $\Phi = TV$ (UDCSMRI(TV)) reconstructed images show clear edge details, the least aliasing and the lowest reconstructed error. Besides, UDCSMRI(TV) reconstructed image obtains the highest PSNR (39.10dB) and lowest RLNE(0.0684). These demonstrate that UDCSMRI performs preeminently in reconstructing T2wBrain_slice27.

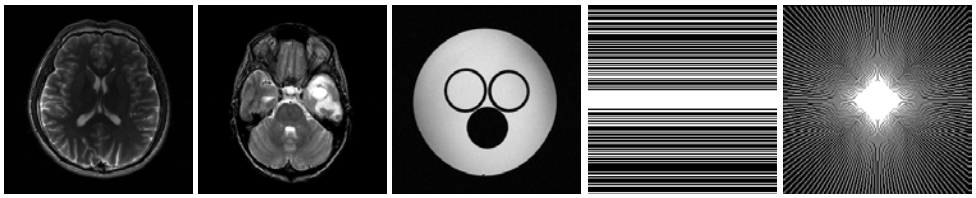
For MBA_T2_slice006 reconstruction under Cartesian sampling scheme at 0.40 sampling rate, the reconstructed images PSNRs of TVCMRI, FCSA, WaTMRI, UDCSMRI(l_1) and UDCSMRI(TV) are 30.15dB, 31.08dB, 30.48dB, 36.01dB and 38.95dB, respectively. RLNEs are 0.1263, 0.1135, 0.1224, 0.0644 and 0.0459 separately. These indicate that UDCSMRI obtains the least reconstruction error.

Water phantom reconstructed results under 30.20% pseudo radial sampling scheme in Fig.4 indicate that TVCMRI, FCSA and WaTMRI can not reduce aliasing efficiently. While UDCSMRI(l_1) and UDCSMRI(TV) reconstructed images obtain clear edge structures. It is worth mentioning that reconstructed result in Fig.4(d) has better rhombic texture features and more clear directions than that in Fig.4(e). It means that UDCSMRI(l_1) performs better than UDCSMRI(TV) in reconstructing water phantom.

AIDS dementia (slice0-16), Brain Tumor (slice0-23) and Normal aging (slice0-53) reconstruction using Cartesian sampling at 0.40 sampling rate are implemented to further test the performance of UDCSMRI. PSNR and RLNE curves versus slices of UDCSMRI reconstruction, for AIDS dementia, Brain Tumor, Normal aging separately, are compared with those of TVCMRI, FCSA, WaTMRI. The comparison curves are exhibited in Fig.5. The statistical results in Fig.5 show that UDCSMRI can reconstruct original MR images from highly undersampled k-space with high probability among all the compared methods.

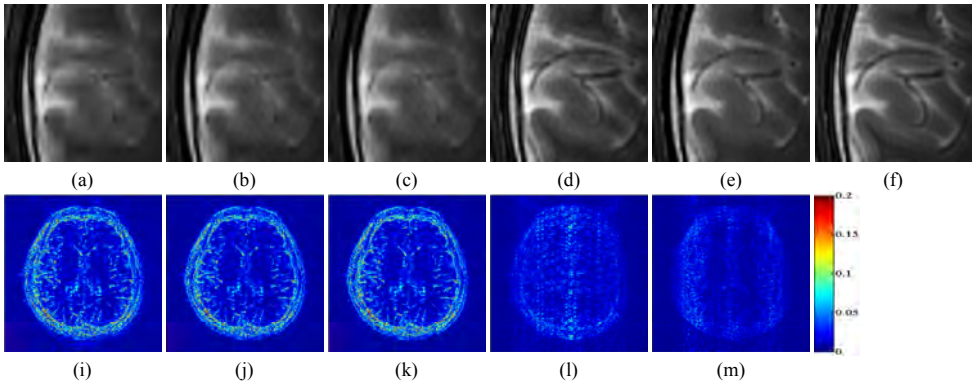
Sampled Data with Noise

The ability of UDCSMRI for handling noise is tested in this subsection. After random gaussian white noise with standard deviation of 10.2 is added to fully sampled k-space data, PSNRs for fully sampled reconstructed T2wBrain_slice27, MBA_T2_slice006 and water phantom are 29.87dB 28.94dB and 30.76dB separately. RLNEs are 0.1980, 0.1451 and 0.0609 separately. Table 1 shows numerical metrics for reconstructed T2wBrain_slice27 and MBA_T2_slice006 using sampling scheme in Fig.2(d) at 0.40 sampling rate, and reconstructed water phantom using sampling scheme in Fig.2(e) at 0.3020 sampling rate, respectively. In Table 1, UDCSMRI reconstructed results obtain the highest PSNR and lowest RLNE, indicating that UDCSMRI can eliminate noise efficiently. TV regularization constrained UDCSMRI performs better that l_1 regularization constrained UDCSMRI in eliminating noise in reconstructing images in Fig.2(a)-(b). While for reconstructing image in Fig.2(c) under



(a) (b) (c) (d) (e)

Figure 2. (a) MR T2wBrain_slice27, (b) MBA_T2_slice006, (c) Water phantom, (d) Cartesian sampling scheme and (e) Pseudo radial sampling scheme.



(a) (b) (c) (d) (e) (f)

(i) (j) (k) (l) (m)

Figure 3. T2wBrain_slice27 reconstruction with Cartesian sampling at 0.40 **sampling rate. (a)-(f) Amplified local regions of reconstructed images from TVCMRI, FCSA, WaTMRI, UDCSMRI(l_1), UDCSMRI(TV) and fully sampled k-space data separately, (g)-(k) Difference image between fully sampled MR image and TVCMRI, FCSA, WaTMRI, UDCSMRI(l_1), UDCSMRI(TV) reconstructed images with gray scale of [**0, 0.20**], respectively. PSNRs of TVCMRI, FCSA, WaTMRI, UDCSMRI(l_1), UDCSMRI(TV) reconstructed images are** 30.74**dB,** 31.29**dB,** 30.87**dB,** 36.41**dB and** 39.10**dB and RLNEs of them are** 0.1790**,** 0.1681**,** 0.1764**,** 0.0932 **and** 0.0684 **separately.**

(a) (b) (c) (d) (e) (f)

Figure 4. Pseudo radial sampling at 0.3020 **sampling rate. (a)-(f) Enlarged local regions of reconstructed water phantom from TVCMRI, FCSA, WaTMRI, UDCSMRI(l_1), UDCSMRI(TV) and fully sampled k-space data separately.**

noise, UDCSMRI(l_1) performs slightly better than UDCSMRI(TV).

Influences of Various Sparse Priors

Influences of various sparse priors to C-SALSA reconstruction performance without noise are discussed in this subsection, for reconstructing T2wBrain_slice27 and MBA_T2_slice006 under Cartesian sampling scheme at 0.40 sampling rate and water phantom under 30.20% pseudo radial sampling scheme. C-SALSA based on Daubechies wavelet basis, less redundant SFLCT(LRSFLCT) based C-SALSA, more redundant SFLCT(MRSFLCT) based C-SALSA, FDCT based C-SALSA and UDCSMRI reconstruction methods are compared in our work. In simulation, regularization parameters of compared methods are manually optimized for maximum PSNRs and minimum RLNEs. Table 2 and Table 3 exhibit reconstructed numerical indices using C-SALSA with $\Phi = l_1$ and $\Phi = TV$ separately. Table 2 exhibits clearly that reconstruction based on conventional sparse methods cannot efficiently eliminate artifacts and aliasing caused by Cartesian undersampling, particularly for wavelet and FDCT based C-SALSA. MRSFLCT based C-SALSA reconstructed images obtain slightly higher PSNRs and lower RLNEs separately than LRSFLCT based C-SALSA reconstructed images, indicating that increasing redundancy properly can improve the reconstruction quality to some extent. While UDCSMRI reconstructed images possess highest PSNRs and lowest RLNEs, indicating that UDCT performs best in sparsifying MR images and thus can lead to lower undersampling rate while

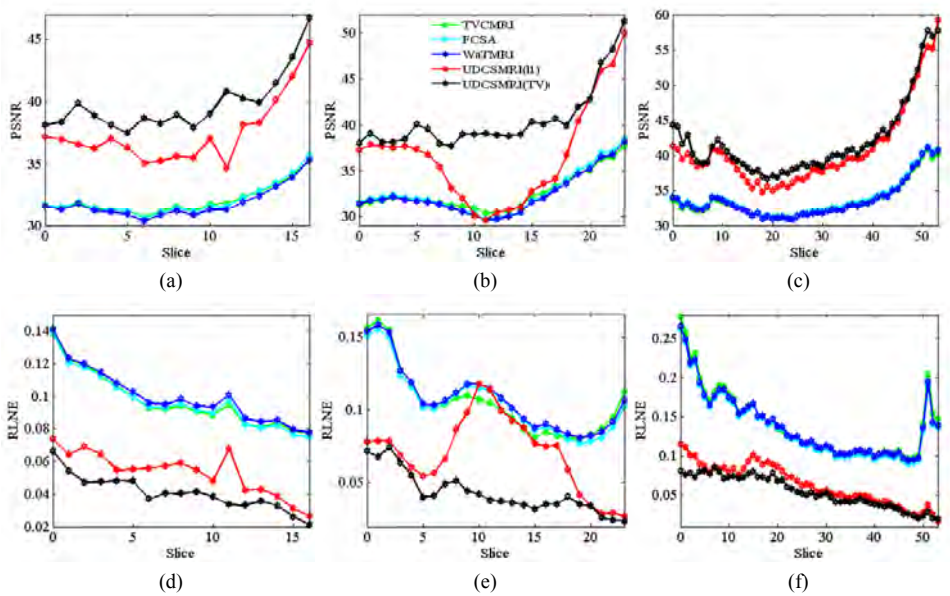


Figure 5. Cartesian sampling at 0.40 **sampling rate. (a)-(c) PSNR versus slices for AIDS dementia, Brain Tumor and Normal aging, respectively. (d)-(f) RLNE versus slices for AIDS dementia, Brain Tumor and Normal aging, respectively.**

Images & Sampling schemes	Indices	Methods				
		TVCMRI	FCSA	WaTMRI	UDCSMRI(l_1)	UDCSMRI(TV)
T2wBrain_slice27 & Cartesian	PSNR(dB)	28.79	28.67	28.38	31.84	**32.24**
	RLNE	0.2241	0.2272	0.2349	0.1577	**0.1507**
MBA_T2_slice006 & Cartesian	PSNR(dB)	29.63	29.57	29.32	31.36	**31.76**
	RLNE	0.1341	0.1351	0.1390	0.1099	**0.1049**
water phantom & pseudo	PSNR(dB)	12.62	9.43	9.38	**33.03**	32.80
	RLNE	0.4917	0.7102	0.7140	**0.0469**	0.0482

Table 1. Reconstructed images quality indices for sampled data with noise

Images & Sampling schemes	Indices	Sparse priors				
		Daubechies wavelet	LRSFLCT	MRSFLCT	FDCT	UDCT
T2wBrain_slice27 & Cartesian	PSNR(dB)	32.91	33.79	34.73	33.34	**36.41**
	RLNE	0.1395	0.1260	0.1131	0.1327	**0.0932**
MBA_T2_slice006 & Cartesian	PSNR(dB)	31.49	31.15	32.19	30.28	**36.01**
	RLNE	0.1083	0.1125	0.0998	0.1245	**0.0644**
water phantom & pseudo	PSNR(dB)	33.86	35.01	35.28	33.88	**35.74**
	RLNE	0.0426	0.0374	0.0362	0.0425	**0.0343**

Table 2. Various sparse priors with l_1 regularization

obtaining high-quality reconstruction. Table 3 shows similar reconstruction results in general. What worth mentioning is that MRSFLCT and LRSFLCT based C-SALSA ($\Phi = TV$) obtain the same numerical indices. Comparing Table 2 with Table 3, it can be concluded that l_1 regularization performs better than TV regularization for sparse transforms except UDCT.

Convergence Analysis

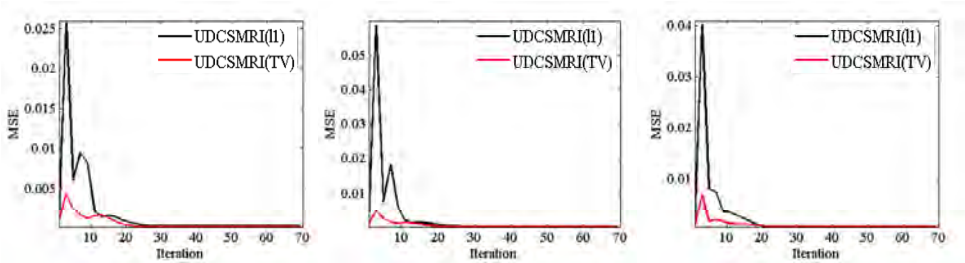
Convergence of UDCSMRI reconstruction is analyzed in this subsection. MSE versus ADMM-2 iteration number for reconstructing Fig.3(d) and (e), MBA_T2_slice006 under the same conditions and Fig.4(d) and (e) are exhibited in Fig.6. When iteration number reaches 25, MSE has already fell into minimal values. Conclusions are made that UDCSMRI(l_1) and UDCSMRI(TV) can obtain rapid convergence with very small MSEs.

4 CONCLUSIONS AND FUTURE WORK

A simple and efficient uniform discrete curvelet transform sparsity based CS MRI framework has been proposed in this paper. In this framework, UDCT obtains optimal structural sparsity, laying the foundation of high quality reconstruction from ill-posed linear in-

Images & Sampling schemes	Indices	Sparse priors				
		Daubechies wavelet	LRSFLCT	MRSFLCT	FDCT	UDCT
T2wBrain_slice27 & Cartesian	PSNR(dB)	28.45	31.40	31.41	30.82	**39.10**
	RLNE	0.2331	0.1659	0.1658	0.1774	**0.0684**
MBA_T2_slice006 & Cartesian	PSNR(dB)	26.80	30.44	30.44	30.08	**38.95**
	RLNE	0.1857	0.1221	0.1221	0.1274	**0.0459**
water phantom & pseudo	PSNR(dB)	31.11	33.01	33.01	33.01	**34.42**
	RLNE	0.0585	0.0470	0.0470	0.0470	**0.0400**

Table 3. Various sparse priors with TV regularization



(a) (b) (c)

Figure 6. MSEs decline versus iteration. (a) Fig.3(d) and (e) reconstruction. (b) MBA_T2_slice006 reconstruction under the same conditions. (c) Fig.4(d) and (e) reconstruction.

verse problems. C-SALSA enforces optimized images transform sparsity and data fidelity at fast convergence speed. Experiments on various MR images illustrate the proposed method can achieve low reconstruction error among current CS MRI methods. The proposed method obtains preeminent reconstruction performance at the cost of doubling the amount of calculation due to handling the real part and imaginary part of complex-valued MR images separately, though. Thus, further improvements on the proposed method are subjects of ongoing research and can be made from the following three aspects: (1) Test and optimize the method on more datasets. (2) Expand the method to 3D dynamic MRI by adding sparsity regularization defined along the temporal axis. (3) Use partially parallel imaging(PPI) to accelerate imaging.

5 ETHICS STATEMENT

Brain image of MBA_T2_slice006, AIDS dementia, Brain Tumor and Normal aging were downloaded from http://www.med.harvard.edu/AANLIB/home.html.
The rest human images were acquired from healthy subjects under the approval of the Institute Review Board of Xiamen University and written consent was obtained from the participants. The data were analyzed anonymously.

6 ACKNOWLEDGMENTS

This work was partially supported by National Natural Science Foundation of China (Grant No.61175012, 61201421, 61201422), Science Foundation of Gansu Province of China (Grant No.1208RJZA265), Specialized Research Fund for the Doctoral Program of Higher Education of China (Grant No.20110211110026) and the Fundamental Research Funds for the Central Universities of China (Grant No.lzujbky-2013-k06, lzujbky-2015-108, lzujbky-2015-197).

7 REFERENCES

[1614066] Donoho, D.L. Compressed sensing. Information Theory, IEEE Transactions on, pp.1289-1306, 2006.

[baraniuk2007compressive] Baraniuk, R. Compressive sensing. IEEE signal processing magazine, 2007.

[lustig2007sparse] Lustig, M., Donoho, D., and Pauly, J.M. Sparse MRI: The application of compressed sensing for rapid MR imaging. Magnetic resonance in medicine, pp.1182-1195, 2007.

[4472246] Lustig, M., Donoho, D.L., Santos, J.M., and Pauly, J.M. Compressed Sensing MRI. Signal Processing Magazine, IEEE, pp.72-82, 2008.

[chen2010novel] Chen, Y.M., Ye, X.J, and Huang, F. A novel method and fast algorithm for MR image reconstruction with significantly under-sampled data. Inverse Problems and Imaging, pp.223-240, 2010.

[santos2006single] Santos, J.M., Cunningham, C.H., Lustig, M., Hargreaves, B.A., Hu, B.S., Nishimura, D.G., and Pauly, J.M. Single breath-hold whole-heart MRA using variable-density spirals at 3t. Magnetic resonance in medicine, pp.371-379, 2006.

[block2007undersampled] Block, K.T., Uecker, M., and Frahm, J. Undersampled radial MRI with multiple coils. Iterative image reconstruction us-

ing a total variation constraint. Magnetic resonance in medicine, pp.1086-1098, 2007.

[haldar2011compressed] Haldar, J.P., Hernando, D., and Liang, Z.P. Compressed-sensing MRI with random encoding. Medical Imaging, IEEE Transactions on, pp.893-903, 2011.

[huang2011efficient] Huang, J.Z., Zhang, S.T., and Metaxas, D. Efficient MR image reconstruction for compressed MR imaging. Medical Image Analysis, pp.670-679, 2011.

[huang2012compressed] Huang, J.Z., and Yang, F. Compressed magnetic resonance imaging based on wavelet sparsity and nonlocal total variation, in Biomedical Imaging (ISBI), 2012 9th IEEE International Symposium on, pp.968-971, 2012.

[1532309] Do, M.N., and Vetterli, M. The contourlet transform: an efficient directional multiresolution image representation. Image Processing, IEEE Transactions on, pp.2091-2106, 2005.

[da2006nonsubsampled] Da, C., Arthur, L., Zhou, J.P., and Do, M.N. The nonsubsampled contourlet transform: theory, design, and applications. Image Processing, IEEE Transactions on, pp.3089-3101, 2006.

[lu2006new] Lu, Y., and Do, M.N. A new contourlet transform with sharp frequency localization, in Image Processing, 2006 IEEE International Conference on, pp.1629-1632, 2006.

[candes2000curvelets] Candes, E.J., and Donoho, D.L. Curvelets: A surprisingly effective nonadaptive representation for objects with edges. DTIC Document, 2000.

[candes2006fast] Candes, E., Demanet, L., Donoho, D., and Ying, L.X. Fast discrete curvelet transforms. Multiscale Modeling & Simulation, pp.861-899, 2006.

[lim2010discrete] Lim, W.Q. The discrete shearlet transform: A new directional transform and compactly supported shearlet frames. Image Processing, IEEE Transactions on, pp.1166-1180, 2010.

[qin2013efficient] Qin, J., and Guo, W.H. An efficient compressive sensing MR image reconstruction scheme, in Biomedical Imaging (ISBI), 2013 IEEE 10th International Symposium on, pp.306-309, 2013.

[jung2009k] Jung, H., Sung, K., Nayak, K.S., Kim, E.Y., and Ye, J.C. k-t FOCUSS: A general compressed sensing framework for high resolution dynamic MRI. Magnetic Resonance in Medicine, pp.103-116, 2009.

[otazo2010combination] Otazo, R., Kim, D., Axel, L., and Sodickson, D.K. Combination of compressed sensing and parallel imaging for highly accelerated first-pass cardiac perfusion MRI. Magnetic Resonance in Medicine, pp.767-776, 2010.

[bilen2012high] Bilen, C., Wang, Y., and Selesnick, I.W. High-speed compressed sensing reconstruction in dynamic parallel MRI using augmented Lagrangian and parallel processing. Emerging and Selected Topics in Circuits and Systems, IEEE Journal on, pp.370-379, 2012.

[qu2010combined] Qu, X.B., Cao, X., Guo, D., Hu, C.W., and Chen, Z. Combined sparsifying transforms for compressed sensing MRI. Electronics letters, pp.121-123, 2010.

[lewicki2000learning] Lewicki, M.S., and Sejnowski, T.J. Learning overcomplete representations. Neural computation, pp.337–365, 2000.

[4494699] Rauhut, H., Schnass, K., and Vandergheynst, P. Compressed Sensing and Redundant Dictionaries. Information Theory, IEEE Transactions on, pp.2210-2219, 2008.

[ning2013magnetic] Ning, B., Qu, X.B., Guo, D., Hu, C.W., and Chen, Z. Magnetic resonance image reconstruction using trained geometric directions in 2D redundant wavelets domain and nonconvex optimization. Magnetic resonance imaging, pp.1611-1622, 2013.

[qu2012undersampled] Qu, X.B., Guo, D., Ning, B.D., Hou, Y.K., Lin, Y.L., Cai, S.H., and Chen, Z. Undersampled MRI reconstruction with patch-based directional wavelets. Magnetic resonance imaging, pp.964-977, 2012.

[dabov2007image] Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K. Image denoising by sparse 3-D transform-domain collaborative filtering. Image Processing, IEEE Transactions on, pp.2080-2095, 2007.

[adluru2010reconstruction] Adluru, G., Tasdizen, T., Schabel, M.C., and DiBella, E.V. Reconstruction of 3D dynamic contrast-enhanced magnetic resonance imaging using nonlocal means. Journal of Magnetic Resonance Imaging, pp.1217-1227, 2010.

[fang2010coherence] Fang, S., Ying, K., Zhao, L., and Cheng, J.P. Coherence regularization for SENSE reconstruction with a nonlocal operator (CORNOL). Magnetic Resonance in Medicine, pp.1413-1425, 2010.

[liang2011sensitivity] Liang, D., Wang, H.F., Chang, Y.C., and Ying, L. Sensitivity encoding reconstruction with nonlocal total variation regularization. Magnetic resonance in medicine, pp.1384-1392, 2011.

[wong2013sparse] Wong, A., Mishra, A., Fieguth, P., and Clausi, D.A. Sparse Reconstruction of Breast MRI Using Homotopic Minimization in a Region-

al Sparsified Domain. Biomedical Engineering, IEEE Transactions on, pp.743-752, 2013.

[akccakaya2011low] Akçakaya, M., Basha, T.A., Goddu, B., Goepfert, L.A., Kissinger, K.V., Tarokh, V., Manning, W.J., and Nezafat, R. Low-dimensional-structure self-learning and thresholding: Regularization beyond compressed sensing for MRI Reconstruction. Magnetic Resonance in Medicine, pp.756-767, 2011.

[qu2014magnetic] Qu, X.B., Hou, Y.K., Lam, F., Guo, D., Zhong, J.H., and Chen, Z. Magnetic resonance image reconstruction from undersampled measurements using a patch-based nonlocal operator. Medical image analysis, pp.843-856, 2014.

[rao1999affine] Rao, B.D., and Kreutz, D.K. An affine scaling methodology for best basis selection. Signal Processing, IEEE Transactions on, pp.187-200, 1999.

[candes2008enhancing] Candes, E.J., Wakin, M.B., and Boyd, S.P. Enhancing sparsity by reweighted l1 minimization. Journal of Fourier analysis and applications, pp.877-905, 2008.

[nocedal2006conjugate] Nocedal, J., and Wright, S.J. Conjugate gradient methods. Springer, 2006.

[aelterman2011augmented] Aelterman, J., Luong, Hiêp.Q., Goossens, B., Pižurica, A., and Philips, W. Augmented Lagrangian based reconstruction of non-uniformly sub-Nyquist sampled MRI data. Signal Processing, pp.2731-2742, 2011.

[van2008probing] Van Den Berg, E., and Friedlander, M.P. Probing the Pareto frontier for basis pursuit solutions. SIAM Journal on Scientific Computing, pp.890-912, 2008.

[yang2011alternating] Yang, J.F., and Zhang, Y. Alternating direction algorithms for l_1-problems in compressive sensing. SIAM journal on scientific computing, pp.250-278, 2011.

[Zhang2010] MATLAB software: http://www.caam.rice.edu/optimization l_1, 2010.

[yang2010fast] Yang, J.F., Zhang, Y., and Yin, W.T. A fast alternating direction method for TVL1-L2 signal reconstruction from partial Fourier data. Selected Topics in Signal Processing, IEEE Journal of, pp.288-297, 2010.

[becker2011nesta] Becker, S., Bobin, Jérôme., Candès, E.J. NESTA: a fast and accurate first-order method for sparse recovery. SIAM Journal on Imaging Sciences, pp.1-39, 2011.

[5570998] Afonso, M.V., Bioucas-Dias, J.M., and Figueiredo, M. A T. An Augmented Lagrangian Approach to the Constrained Optimization Formulation of Imaging Inverse Problems. Image Processing, IEEE Transactions on, pp.681-695, 2011.

[5443489] Nguyen, T.T., and Chauris, H. Uniform Discrete Curvelet Transform. Signal Processing, IEEE Transactions on, pp.3618-3634, 2010.

[esser2009applications] Esser, E. Applications of Lagrangian-based alternating direction methods and connections to split Bregman. CAM report, pp.31, 2009.

[natarajan1995sparse] Natarajan, B.K. Sparse approximate solutions to linear systems. SIAM journal on computing, pp.227-234, 1995.

[4303060] Chartrand, R. Exact Reconstruction of Sparse Signals via Nonconvex Minimization. Signal Processing Letters, IEEE, pp.707-710, 2007.

[ma2008efficient] Ma, S.Q., Yin, W.T., Zhang, Y., and Chakraborty, A. An efficient algorithm for compressed MR imaging using total variation and wavelets, in Computer Vision and Pattern Recognition, CVPR 2008, IEEE Conference on, pp.1-8, 2008.

[NIPS20124630] Chen C., and Huang, J.Z. Compressive Sensing MRI with Wavelet Tree Sparsity. Advances in Neural Information Processing Systems 25, pp.1115-1123, 2012.

[shin2013calibrationless] Shin, P.J., Larson, P.EZ., Ohliger, M.A., Elad, M., Pauly, J.M., Vigneron, D.B., and Lustig, M. Calibrationless parallel imaging reconstruction based on structured low-rank matrix completion. Magnetic Resonance in Medicine, 2013.

[chambolle2004algorithm] Chambolle, A. An algorithm for total variation minimization and applications. Journal of Mathematical imaging and vision, pp.89-97, 2004.

[Quwebsite2010] http://www.quxiaobo.org/index_publications.html, 2010.

[qu2002information] Qu, G.H., Zhang, D.L., and Yan, P.F. Information measure for performance of image fusion. Electronics letters, pp.313–315, 2002.

[liang2009accelerating] Liang, D., Liu, B., Wang, J.J., and Ying,L. Accelerating SENSE using compressed sensing. Magnetic Resonance in Medicine, pp.1574–1584, 2009.

[afonso2010fast] Afonso, M.V., Bioucas-Dias, José.M., and Figueiredo, Mário.AT. Fast image recovery using variable splitting and constrained optimization. Image Processing, IEEE Transactions on, pp.2345–2356, 2010.

Multiframe Visual-Inertial Blur Estimation and Removal for Unmodified Smartphones

Gábor Sörös, Severin Münger, Carlo Beltrame, Luc Humair
Department of Computer Science
ETH Zurich
soeroesg@ethz.ch, {muengers|becarlo|humairl}@student.ethz.ch

ABSTRACT

Pictures and videos taken with smartphone cameras often suffer from motion blur due to handshake during the exposure time. Recovering a sharp frame from a blurry one is an ill-posed problem but in smartphone applications additional cues can aid the solution. We propose a blur removal algorithm that exploits information from subsequent camera frames and the built-in inertial sensors of an unmodified smartphone. We extend the fast non-blind uniform blur removal algorithm of Krishnan and Fergus to non-uniform blur and to multiple input frames. We estimate piecewise uniform blur kernels from the gyroscope measurements of the smartphone and we adaptively steer our multiframe deconvolution framework towards the sharpest input patches. We show in qualitative experiments that our algorithm can remove synthetic and real blur from individual frames of a degraded image sequence within a few seconds.

Keywords

multiframe blur removal, deblurring, smartphone, camera, gyroscope, motion blur, image restoration

1 INTRODUCTION

Casually taking photographs or videos with smartphones has become both easy and widespread. There are, however, two important effects that degrade the quality of smartphone images. First, handshake during the exposure is almost unavoidable with lightweight cameras and often results in motion-blurred images. Second, the rolling shutter in CMOS image sensors introduces a small time delay in capturing different rows of the image that causes image distortions. Retaking the pictures is often not possible, hence there is need for post-shoot solutions that can recover a sharp image of the scene from the degraded one(s). In this paper, we address the problem of blur removal and rolling shutter rectification for *unmodified* smartphones, i.e., without external hardware and without access to low-level camera controls.

In the recent years, a large number of algorithms have been proposed for restoring blurred images. As blur (in the simplest case) is modeled by a convolution of a sharp image with a blur kernel, blur removal is also termed deconvolution in the literature. We distinguish between non-blind deconvolution, where the blur kernel is known in advance, and blind deconvolution, where the blur kernel is unknown and needs to be estimated first. The blur kernel can be estimated for instance from salient edges in the image [Jos08, Cho09, Sun13, Xu13], from the frequency domain [Gol12], from an auxiliary camera [Tai10], or from motion sensors [Jos10]. Kernel estimation from the image content alone often involves iterative optimization schemes that are computationally too complex to perform on a smartphone within acceptable time. Auxiliary hardware might be expensive and difficult to mount, so kernel estimation from built-in motion sensors seems the most appealing for smartphone applications.

Unfortunately, even known blur is difficult to invert because deconvolution is mathematically ill-posed which means many false images can also satisfy the equations. Deconvolution algorithms usually constrain the solution space to images that follow certain properties of natural images [Kri09]. Another common assumption is uniform blur over the image which simplifies the mathematical models and allows for faster restoration algorithms. However, this is usually violated in real scenarios which can lead the restoration to fail, often even lowering the quality of the processed image [Lev09]. Handling different blur at each pixel of the image is computationally demanding, so for smartphone applications a semi-non-uniform approach might be the best that divides the image to smaller overlapping regions, where uniform blur can be assumed, and restores those regions independently.

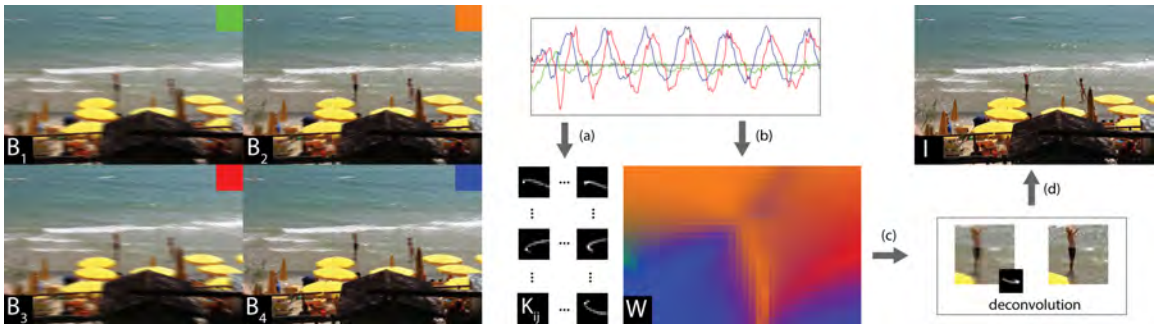


Figure 1: Illustration of our gyroscope-aided multiframe blur removal algorithm. *(a)* First, piecewise uniform blur kernels $\mathbf{K_{ij}}$ along rows i and columns j of the image are estimated from the gyroscope measurements. *(b)* Next, a blurriness map $\mathbf{W}$ is generated by measuring the spatial extent of the respective kernels. *(c)* Then, individual patches are restored from multiple blurry input patches of $\mathbf{B_i}$ using natural image priors. *(d)* Finally, the sharp output $\mathbf{I}$ is assembled from the deblurred patches.

We build our blur removal algorithm on the following three observations: 1) The blurry image of a point light source in the scene (such as distant street lights at night) gives the motion blur kernel at that point of the image. 2) Smartphones today also provide a variety of sensors such as gyroscopes and accelerometers that allow reconstructing the full camera motion during camera exposure thereby giving information about the blur process. 3) Information from multiple degraded images of the same scene allows restoring a sharper image with more visible details.

Contributions

Based on the above observations we propose a new fast blur removal algorithm for unmodified smartphones by using subsequent frames from the camera preview stream combined with the built-in inertial sensors (see Figure 1). We assume a static scene and that blur is mainly caused by rotational motion of the camera. The motion of the camera during the shot is reconstructed from the gyroscope measurements, which requires time synchronization of the camera and the inertial sensors. The information from multiple subsequent frames with different blur is exploited to reconstruct a sharp image of the scene. To the best of our knowledge, this is the first application that combines blur kernel estimation from inertial sensors, patch-based non-uniform deblurring, multiframe deconvolution with natural image priors, and correction of rolling shutter deformation for unmodified smartphones. The runtime of our algorithm is in the order of a few seconds for typical preview frame size of 720×480 pixels.

2 RELATED WORK

The use of multiple blurry/noisy images for restoring a single sharp frame (without utilizing sensor data) has been proposed by Rav-Acha and Peleg [Rav05], applied for sharp panorama generation by Lie et al. [Li10], and recently for HDR and low-light photography by Ito et al [Ito14]. Combining camera frames and inertial measurement units (IMU) has been successfully used for video stabilization [For10, Han11, Kar11, Bel14], for denoising [Ito14, Rin14] and also for blur removal [Jos10, Par14, Sin14b].

Joshi et al. [Jos10] presented the first IMU-based deblurring algorithm with a DSLR camera and external gyroscope and accelerometer. In their approach, the camera and the sensors are precisely synchronized through the flash trigger. The DSLR camera has a global shutter which makes the problem easier to handle than the case of smartphones. They assume a constant uniform scene depth, which they find together with the sharp image by solving a complex optimization problem. Bae et al. [Bae13] extend this method to depth-dependent blur by attaching a depth sensor to the camera. Ringaby and Forssen [Rin14] develop a virtual tripod for smartphones by taking a series of noisy photographs, aligning them using gyroscope data, and averaging them to get a clear image, but not targeting blur removal. Köhler [Koh12] and Whyte [Why12] show in their single-frame deblurring experiments that three rotations are enough to model real camera shakes well.

Karpenko [Kar11], Ringaby [For10], and Bell [Bel14] developed methods for video stabilization and rolling shutter correction specifically for smartphones. An important issue in smartphone video stabilization is the synchronization of the camera and the IMU data because the mobile operating systems do not provide precise timestamps. Existing methods estimate the unknown parameters (time delay, frame rate, rolling shutter fill time, gyroscope drift) from a sequence of images off-line via optimization. We have found that the camera parameters might even change over time, for example the smartphones automatically adjust the frame rate depending on whether we capture a bright or a dark scene. This is an important issue because

it means we require an online calibration method. Jia and Evans [Jia14] proposed such an online camera-gyroscope calibration method for smartphones based on an extended Kalman filter (EKF). The method tracks point features over a sequence of frames and estimates the time delay, the rolling shutter fill rate, the gyroscope drift, the physical sensor offset, and even the camera intrinsics. However, it requires clean frames for feature tracking.

Recently, Park and Levoy [Par14] compared the effectiveness of jointly deconvolving multiple images degraded by small blur versus deconvolving a single image degraded by large blur, and versus averaging a set of blur-free but noisy images. They record a series of images together with gyroscope measurements on a modified tablet with advanced camera controls (e.g., exposure control, RAW data access) through the FCam API [Par11], and attach an external 750Hz gyroscope to the tablet. They estimate the rotation axis, the gyroscope drift, and the time delay between the frames and the gyroscope measurements in a non-linear optimization scheme over multiple image segments. Their optimization scheme is based on the fact that applying two different kernels to an image patch is commutative. This means in the case of true parameters, applying the generated kernels in different order results in the same blurry patch. The rolling shutter parameter is calculated off-line with the method of Karpenko [Kar11]. They report the runtime of the algorithm to be 24.5 seconds using the Wiener filter and 20 minutes using a sparsity prior for deconvolution, not mentioning whether on the tablet or on a PC.

Closest to our system is the series of work by Sindelar et al. [Sin13, Sin14a, Sin14b] who also reconstruct the blur kernels from the sensor readings of an unmodified smartphone in order to make the deconvolution non-blind. The first approach [Sin13] considers only x and y rotations and generates a single kernel as weighted line segments using Bresenham's line drawing algorithm. Unfortunately, their time calibration method is not portable to different phones. They find the exact beginning of the exposure by inspecting the logging output of the camera driver, which might be different for each smartphone model. They read the exposure time from the EXIF tag of the captured photo, however, this information is not available for the preview frames we intend to use for a live deblurring system. Extending this work in [Sin14b] the authors generate piecewise uniform blur kernels and deblur overlapping patches of the image using the Wiener filter. They also account for rolling shutter by shifting the time window of gyroscope measurements when generating blur kernels for different rows of the image. The main difference in our approach is the use of multiple subsequent images and non-blind deblurring with natural image priors.

3 BLUR MODEL

The traditional convolution model for uniform blur is written in matrix-vector form as

$$\vec{\mathbf{B}} = \mathbf{A}\vec{\mathbf{I}} + \vec{\mathbf{N}} \tag{1}$$

where $\vec{\mathbf{B}}$,$\vec{\mathbf{I}}$,$\vec{\mathbf{N}}$ denote the vectorized blurry image, sharp image, and noise term, respectively, and $\mathbf{A}$ is the sparse blur matrix. Camera shake causes non-uniform blur over the image, i.e., different parts of the image are blurred differently. We assume piecewise uniform blur and use different uniform kernels for each image region which is a good compromise between model accuracy and model complexity.

The blur kernels across the image can be found by reconstructing the camera movement, which is a path in the six-dimensional space of 3 rotations and 3 translations. A point in this space corresponds to a particular camera pose, and a trajectory in this space corresponds to the camera movement during the exposure. From the motion of the camera and the depth of the scene, the blur kernel at any image point can be derived. In this paper, we target unmodified smartphones without depth sensors so we need to make further assumptions about the scene and the motion.

Similar to Joshi et al. [Jos10], we assume the scene to be planar (or sufficiently far away from the camera) so that the blurred image can be modeled as a sum of transformations of a sharp image. The transformations of a planar scene due to camera movements can be described by a time-dependent homography matrix $\mathbf{H_t} \in \mathbb{R}^{3\times 3}$. We apply the pinhole camera model with square pixels and zero skew for which the intrinsics matrix $\mathbf{K}$ contains the focal length f and the principal point $[c_x, c_y]^T$ of the camera.

Given the rotation matrix $\mathbf{R}_t$ and the translation vector $\mathbf{T_t}$ of the camera at a given time t, the homography matrix is defined as

$$\mathbf{H}_t(d) = \mathbf{K}(\mathbf{R}_t + \frac{1}{d}\mathbf{T}_t\vec{n}^T)\mathbf{K}^{-1} \tag{2}$$

where $\vec{n}$ is the normal vector of the latent image plane and d is the distance of the image plane from the camera center. The homography $\mathbf{H}_t(d)$ maps homogenous pixel coordinates from the latent image $\mathbf{I}_0$ to the transformed image $\mathbf{I}_t$ at time t:

$$\mathbf{I}_t\left((u_t, v_t, 1)^T\right) = \mathbf{I}_0\left(\mathbf{H}_t(d)(u_0, v_0, 1)^T\right) \tag{3}$$

The transformed coordinates in general are not integer valued, so the pixel value has to be calculated via bilinear interpolation, which can also be rewritten as a matrix multiplication of a sparse sampling matrix $\mathbf{A}_t(d)$ with the latent sharp image $\vec{\mathbf{I}}_0$ as $\vec{\mathbf{I}}_t = \mathbf{A}_t(d)\vec{\mathbf{I}}_0$ in vector

form. Then, we can describe the blurry image as the integration of all transformed images during the exposure time plus noise:

$$\vec{\mathbf{B}} = \int_{t_{open}}^{t_{close}} \mathbf{A}_t \cdot \vec{\mathbf{I}} dt + \vec{\mathbf{N}} = \mathbf{A} \cdot \vec{\mathbf{I}} + \vec{\mathbf{N}} \tag{4}$$

Note how this expression resembles the form of (1). While for this formula the depth of the scene is required, a common simplification is to assume zero translation [Kar11, Han11, For10] because rotation has a significantly larger impact on shake blur [Koh12, Why12, Bel14]. With only rotational motion, equation 2 is no longer dependent on the depth:

$$\mathbf{H}_t = \mathbf{K}\mathbf{R}_t\mathbf{K}^{-1} \tag{5}$$

There are also other reasons why we consider only rotational motion in our application. The smartphone's accelerometer measurements include gravity and are contaminated by noise. The acceleration values need to be integrated twice to get the translation of the camera and so the amplified noise may lead to large errors in kernel estimation.

Handling pixel-wise spatially varying blur is computationally too complex to perform on a smartphone, so we adopt a semi-non-uniform approach. We split the images into $R \times C$ overlapping regions (R and C are chosen so that we have regions of size 30×30 pixels) where we assume uniform blur and handle these regions separately. We reconstruct the motion of the camera during the exposure time from the gyroscope measurements and from the motion we reconstruct the blur kernels for each image region by transforming the image of a point light source with the above formulas. Once we have the blur kernels, fast uniform deblurring algorithms can be applied in each region, and the final result can be reassembled from the deblurred regions (possibly originating from different input frames). An overview of our whole pipeline is illustrated in Figure 2.

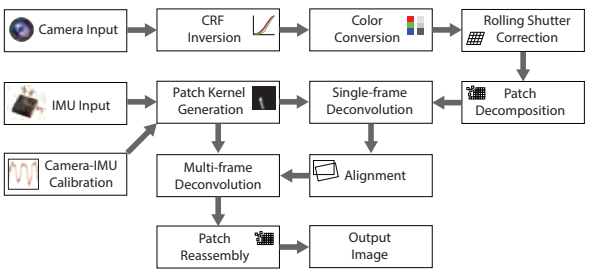


Figure 2: Overview of our blur removal pipeline

3.1 Camera-IMU calibration

Reconstructing the motion of the mobile phone during camera exposure of a given frame i with timestamp t_i requires the exact time window of sensor measurements during that frame. This is challenging to find on unmodified smartphones given that current smartphone APIs allow rather limited hardware control. We denote with t_d the delay between the recorded timestamps of sensor measurements and camera frames which we estimate prior to deblurring.

In rolling shutter (RS) cameras, the pixel values are read out row-wise from top to bottom which means 'top' pixels in an image will be transformed with 'earlier' motion than 'bottom' pixels, which has to be taken into account in our model. For an image pixel $u = [u_x, u_y]^T$ in frame i the start of the exposure is modeled as

$$t([u_x, u_y]^T, i) = t_i + t_d + t_r \frac{u_y}{h} \tag{6}$$

where t_r is the readout time for one frame and h is the total number of rows in one frame. The gyro-camera delay t_d is estimated for the first row of the frame, and the other rows are shifted in time within the range $[0, t_r]$. We set the time of each image region to the time of the center pixel in the region.

To find the unknown constants of our model, we apply once the Extended Kalman Filter (EKF)-based online gyro-camera calibration method of Jia and Evans [Jia13, Jia14]. This method estimates the rolling shutter parameter t_r, the camera intrinsics f, c_x, c_y, the relative orientation of the camera and IMU, the gyroscope bias, and even the time delay t_d. The intrinsics do not change in our application, and the relative orientation is not important because we are only interested in rotation changes which are the same in both coordinate systems. The gyroscope bias is a small and varying additive term on the measured rotational velocities which slightly influences the kernel estimation when integrated over time. However, for kernel estimation we consider only rotation changes during single camera frames, and in such short time intervals the effect of the bias can be neglected. For example, in case of a 30 Hz camera and a 200 Hz gyroscope we integrate only $200/30 \approx 6$ values during a single frame. We perform the online calibration once at the beginning of our image sequences and we assume the above parameters to be constant for the time of capturing the frames we intend to deblur. The EKF is initialized with the intrinsic values given by the camera calibration method in OpenCV.

The time delay t_d was found to slightly vary over longer sequences, so after an initial guess from the EKF, we continuously re-estimate t_d in a background thread. We continuously calculate the mean pixel shift induced by the movement measured by the gyroscope, and we also observe the mean pixel shifts in the images. The current t_d is found by correlating the curves in a sliding time window.

The last parameter, the exposure time $t_e = t_{close} - t_{open}$ of a frame can be read from the EXIF tag of JPEG images like in [Sin13], but an EXIF tag is not available for live video frames. Therefore, we lock the camera settings at the beginning and capture the first frame as a single image.

3.2 Kernel estimation from gyroscope measurements

We generate a synthetic blur kernel at a given point in the image by replaying the camera motion with a virtual camera that is looking at a virtual point light source. For any pixel $u = [u_x, u_y]$ in the image, we place the point light source to $U = [(u_x - c_x)\frac{d}{f}, (u_y - c_y)\frac{d}{f}, d]$ in 3D space. Note that the value of d is irrelevant if we consider rotations only.

First, we need to rotate all sensor samples into a common coordinate system because the raw values are measured relative to the current camera pose. The chosen reference pose is the pose at the shutter opening. Next, the measured angular velocities need to be integrated to get rotations. As described in section 3.1, we neglect the effects of gyroscope bias within the short time of the exposure. In order to get a continuous rendered kernel, we super-resolve the time between discrete camera poses where measurements exist, using spherical linear interpolation (SLERP). The transformed images of the point light source are blended together with bilinear interpolation and the resulting kernel is normalized to sum to 1. Finally, we crop the kernel to its bounding square in order to reduce the computational effort in the later deblurring step.

3.3 Camera response function

The non-linear camera response function (CRF) that converts the scene irradiance to pixel intensities has a significant impact on deblurring algorithms [Tai13]. The reason why manufacturers apply a non-linear function is to compensate the non-linearities of the human eye and to enhance the look of the image. The CRF is different for each camera model and even for different capture modes of the same camera [Xio12]. Some manufacturers disclose their CRFs but for the wide variety of smartphones only few data is available. The CRF of digital cameras can be calibrated for example using exposure bracketing [Deb97] but the current widespread smartphone APIs do not allow exposure control. The iOS 6, the upcoming Android 5.0, and custom APIs such as the FCam [Par11] expose more control over the camera but are only available for a very limited set of devices. To overcome this limitation, we follow the approach of [Li10] and assume the CRF to be a simple gamma curve with exponent 2.2. While this indeed improves the quality of our deblurred images, an online photometric calibration algorithm that tracks the automatic capture settings remains an open question.

4 SINGLE-FRAME BLUR REMOVAL

Given the piecewise uniform blur kernels, we apply the fast non-blind uniform deconvolution method of Krishnan and Fergus [Kri09] on each individual input patch to produce sharper estimates (see Figure 3). The algorithm enforces a hyper-Laplacian distribution on the gradients in the sharp image, which has been shown to be a good natural image prior [Lev09]. Assuming the image has N pixels in total, the algorithm solves for the image $\mathbf{I}$ that minimizes the following energy function:

$$\arg\min_{\mathbf{I}} \sum_{i=1}^{N} \frac{\lambda}{2}(\mathbf{I} * k - \mathbf{B})_i^2 + |(\mathbf{I} * f_x)_i|^\alpha + |(\mathbf{I} * f_y)_i|^\alpha \quad (7)$$

where k is the kernel, and $f_x = [1\ -1]$ and $f_y = [1\ -1]^T$ denote differential operators in horizontal and vertical direction, respectively. λ is a balance factor between the data and the prior terms. The notation $F_i^d\mathbf{I} := (\mathbf{I} * f_d)_i$ and $K_i\mathbf{I} := (\mathbf{I} * k)_i$ will be used in the following for brevity. Introducing auxiliary variables w_i^x and w_i^y (together denoted as $\mathbf{w}$) at each pixel i allows moving the $F_i^d\mathbf{I}$ terms outside the $|\cdot|^\alpha$ expression, thereby separating the problem into two sub-problems.

$$\begin{aligned}\arg\min_{\mathbf{I},\mathbf{w}} \sum_{i=1}^{N} \frac{\lambda}{2}(K_i\mathbf{I} - \mathbf{B}_i)^2 + \frac{\beta}{2}\|F_i^x\mathbf{I} - w_i^x\|_2^2 \ + \\ + \ \frac{\beta}{2}\|F_i^y\mathbf{I} - w_y^2\|_2^2 + |w_i^x|^\alpha + |w_i^y|^\alpha\end{aligned} \quad (8)$$

The β parameter enforces the solution of eq. 8 to converge to the solution of eq. 7, and its value is increased in every iteration. Minimizing eq. 8 for a fixed β can be done by alternating between solving for $\mathbf{I}$ with fixed $\mathbf{w}$ and solving for $\mathbf{w}$ with fixed $\mathbf{I}$. The first sub-problem is quadratic, which makes it simple to solve in the Fourier domain. The second sub-problem is pixel-wise separable, which is trivial to parallelize. Additionally, for certain values of α an analytical solution of the w-subproblem can be found, especially for $\alpha = \frac{1}{2}, \frac{2}{3}, 1$, and for other values a Newton-Raphson root finder can be applied. We experimentally found that $\alpha = \frac{1}{2}$ gives the best results. For further details of the algorithm please refer to [Kri09]. For smoothing discontinuities that may produce ringing artifacts, we perform edge tapering on the overlapping regions before deblurring.

5 MULTI-FRAME BLUR REMOVAL

One of the main novelties of our method is to aid the restoration of a single frame $\mathbf{B}$ with information from preceding and/or subsequent degraded frames $\mathbf{B}_j, 1 \leq j \leq M$ from the camera stream. We first undistort each input frame using the RS-rectification method of

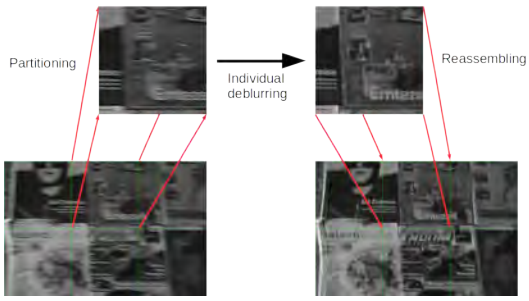


Figure 3: Illustration of our piecewise uniform deblurring method with $R = 2$ rows and $C = 3$ columns. The regions are deblurred independently with the respective kernels and afterwards reassembled in the final image.

Karpenko et al. [Kar11] which is a simple texture warping step on the GPU. We initialize the warping with the same gyroscope data that is used for kernel estimation. Next, we perform single-frame deblurring on every $\mathbf{B}_j$ as well as $\mathbf{B}$ to get sharper estimates $\tilde{\mathbf{I}}_j$ and $\tilde{\mathbf{I}}$, respectively.

After deblurring the single frames, we align all sharpened estimates with $\tilde{\mathbf{I}}$. For each $\tilde{\mathbf{I}}_j$, we calculate the planar homography $\mathbf{H}_j$ that maps $\tilde{\mathbf{I}}_j$ to $\tilde{\mathbf{I}}$. To find the homographies, we perform basic SURF [Bay08] feature matching between each estimate $\tilde{\mathbf{I}}_j$ and $\tilde{\mathbf{I}}$ in a RANSAC loop. Each homography is calculated from the inlier matches such that the reprojection error is minimized. We found that this method is robust even in the case of poorly restored estimates (e.g., in case of large blurs); however, the homography matching can fail if there are many similar features in the recorded scene. Frames that fail the alignment step are discarded.

Finally, we patch-wise apply the warped estimates $\mathbf{I}_j$ as additional constraints on $\mathbf{I}$ in our modified deblurring algorithm. We add a new penalty term γ_{multi} to equation 7 which describes the deviation of the latent image $\mathbf{I}$ from the M other estimates:

$$\gamma_{multi} = \frac{\mu}{2\sum_{j=1}^{M}\mu_j}\sum_{j=1}^{M}\mu_j||\mathbf{I}-\mathbf{I}_j||_2^2 \qquad (9)$$

The weights μ_j are chosen inversely proportional to the 'blurriness' of the corresponding patch in image $\mathbf{B}_j$. The blurriness is defined as the standard deviation (spatial extent) of the blur kernel in the patch. Note that the weights are recalculated for every patch in our non-uniform deblurring approach. The benefit of calculating the weights for each patch independently from the rest of the image is that we can both *spatially* and *temporally* give more weight to sharper patches in the input stack of images. This is advantageous if the same part of the scene got blurred differently in subsequent images. An example colormap of the weights (W) is visualized in Figure 1 where each input image is represented as a distinct color and W shows how much the different input patches influence an output tile.

Analog to the single-frame case, we proceed with the half-quadratic penalty method to separate the problem into $\mathbf{I}$ and $\mathbf{w}$ sub-problems. In the extended algorithm, we only need to calculate one additional Fourier transform in each iteration which keeps our multi-frame method fast. The step-by-step derivation of the solution can be found in the supplemental material.

6 EXPERIMENTS

We have implemented the proposed algorithm in OpenCV[1] and the warping in OpenGL ES 2.0[2] which makes it portable between a PC and a smartphone. We recorded grayscale camera preview sequences of resolution 720×480 at 30 Hz together with gyroscope data at 200 Hz on a Google Nexus 4 smartphone and we performed the following experiments on a PC. The gyro-camera calibration is performed on the first few hundred frames with Jia's implementation [Jia14] in Matlab.

6.1 Kernel generation

To test the accuracy of kernel generation, we recorded a sequence in front of a point light grid, and also generated the kernels from the corresponding gyroscope data. Ideally, the recorded image and the generated image should look identical, and after (single-frame) deblurring using the generated kernels the resulting image should show a point light grid again. Figure 4 illustrates that our kernel estimates are close to the true kernels, however, the bottom part is not matching perfectly because of residual errors in the online calibration.

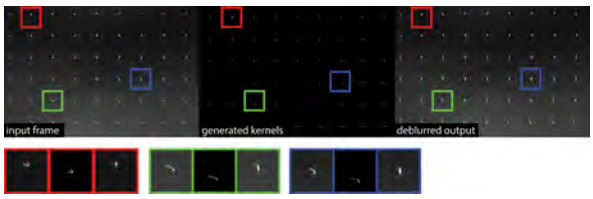


Figure 4: Kernel generation example. Left: blurry photo of a point grid. Middle: kernels estimated from gyroscope data. Right: the deblurred image is close to a point grid again. (Please zoom in for viewing)

6.2 Removing synthetic blur

Starting from a sharp image and previously recorded gyroscope measurements of deliberate handshake, we generated a set of 5 blurry images as input. In our restoration algorithm, we split the 720×480 input images to a grid of $R \times C = 24 \times 16$ regions and for each

[1] http://www.opencv.org/
[2] https://www.khronos.org/opengles/

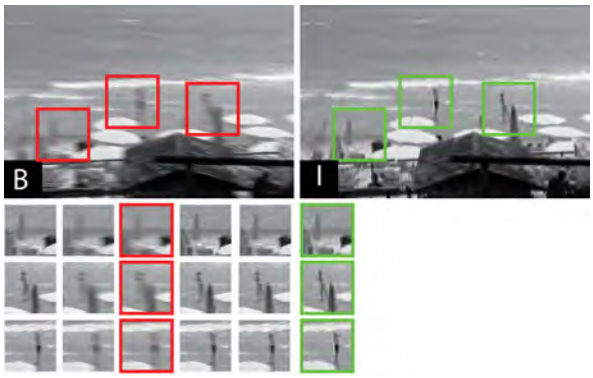


Figure 5: Removing synthetic blur. B is the main input image and 4 neighboring images aid the blur removal, I is our result. Bottom: 3 corresponding patches from the 5 input images, and from the result.

region we individually generate the blur kernel from the gyroscope data. Our multi-frame deconvolution result is shown in Figure 5. The runtime of our algorithm on 5 input images is 18 seconds on a laptop with a 2.40 GHz Core i7-4700MQ CPU (without calibration time).

Next, we test different deconvolution algorithms in our piecewise uniform blur removal for restoring a single frame. We test the Wiener filter, the Richardson-Lucy algorithm, and Krishnan's algorithm as our deconvolution step. For comparison, we also show the results of Photoshop's ShakeReduction feature which is a uniform blind deconvolution method (i.e., can not make use of our gyro-generated kernels). The quality metrics PSNR (peak signal to noise ratio) and SSIM (structure similarity index [Wan04]) of the images are listed in Table 1. The Wiener filter (W) is the fastest method for

	B	W	RL	KS	KM	PS
PSNR	22.374	20.613	23.207	22.999	**24.215**	21.914
SSIM	0.616	0.534	0.657	0.621	**0.673**	0.603

Table 1: Quantitative comparison of various deconvolution steps in our framework. Blurry input image (B), Wiener filter (W), Richardson-Lucy (RL), single-frame Krishnan (KS), multi-frame Krishnan (KM) (3 frames), Photoshop ShakeReduction (PS) (blind uniform deconvolution)

patch-wise single-frame deblurring but produces ringing artifacts that even lower the quality metrics. The output of the Richardson-Lucy algorithm (RL) achieved higher score but surprisingly remained blurry, while Krishnan's algorithm (KS) tends to smooth the image too much. We think this might stem from the fact that the small 30×30 regions do not contain enough image gradients to steer the algorithm to the correct solution. However, Krishnan's algorithm with our extension to multiple input regions (KM) performs the best. Photoshop's ShakeReduction algorithm assumes uniform blur [Cho09] so while it restored the bottom part of the image correctly, the people in the middle of the image remained blurry. The images in higher resolution can be found in the supplement.

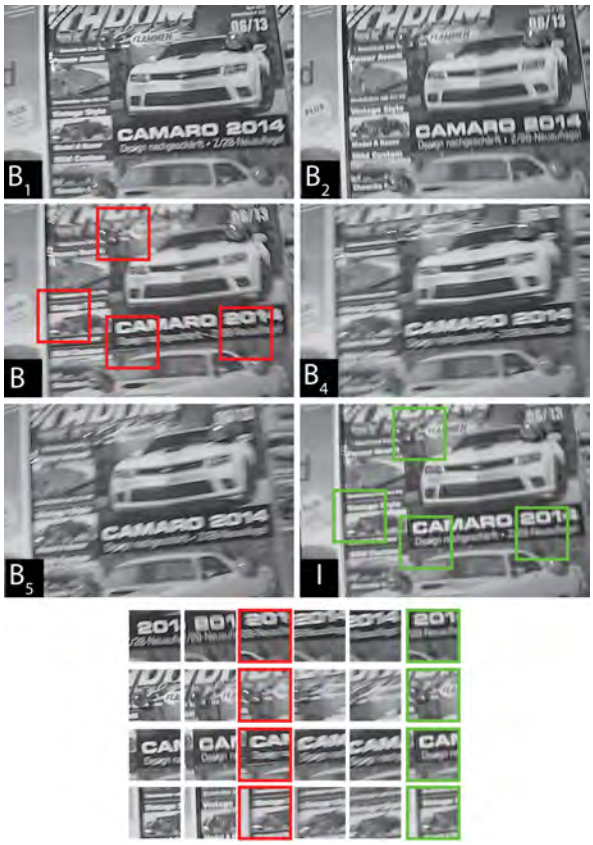


Figure 6: Restoring B with the help of $B_{1,2,4,5}$, all degraded with real motion blur. We also added four marble balls to the scene that act as point light sources and show the true blur kernels at their locations.

6.3 Removing real blur

Figure 6 shows the results of a real example with a static planar scene close to the camera. We used 5 input images degraded by real motion blur. Selected patches illustrate how our algorithm restores different parts of the image by locally steering the deconvolution towards the sharper input tiles. Note, however, that in the second selected patch some details are missing that were present in the sharpest of the input patches. This is because our algorithm does not directly copy that patch from B_1 but applies it within the deconvolution of the corresponding patch in B. A slight misalignment of the five input tiles leads to smoother edges in the multi-frame deconvolution result. The misalignment may stem from the homography estimation which is prone to errors if the single-frame deconvolution is imperfect or from the rolling shutter rectification which is only an approximation of the true image distortion. Figure 7 shows another example of restoring a sharp image from three blurry ones.

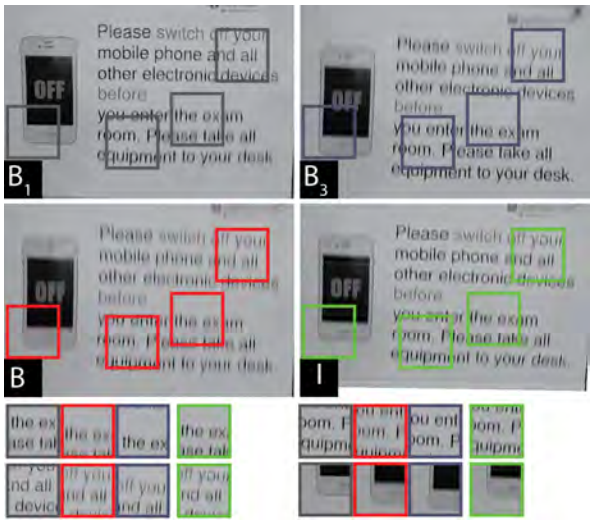


Figure 7: Restoring B with the help of $B_{1,3}$, all degraded with real motion blur. The original misalignment is kept and the rectified result is left uncropped for visualization. (High-resolution images are in the supplement)

6.4 Discussion and limitations

As shown in the experiments, the proposed algorithm is able to restore non-uniformly blurred images. However, there are limitations and assumptions we need to keep in mind. Our non-blind deconvolution step assumes a perfect kernel estimate so a good calibration is crucial for success. The selected gyro-camera calibration method is sensitive to the initialization values which are not trivial to find for different smartphone models. We need to assume that a short sequence with detectable feature tracks for calibration before deblurring exists. However, the calibration does not need to be done every time but only when the camera settings change. We expect that future camera and sensor APIs like StreamInput[3] will provide better synchronization capabilities that allow precise calibration.

Our blur model can generate rotational motion blur kernels at any point of the image, but the model is incorrect if the camera undergoes significant translation or if objects are moving in the scene during the exposure time. In multiframe deblurring, the image alignment based on feature matching might fail if the initial deblurring results are wrong. As we also know the camera motion between the frames, image alignment could be done with the aid of sensor data instead of pure feature matching but then the gyroscope bias needs to be compensated. The restriction to planar scenes and rotational motion overcomes the necessity of estimating the depth of the scene at each pixel.

Our algorithm was formulated for grayscale images. Extending it to color images would be possible by solving the grayscale problem for the RGB color channels separately, however, the color optimizations of the smartphone driver may introduce different non-linear CRFs for each channel, which needs to be handled carefully. The calibration of the per-channel CRFs using standard methods will become possible with the upcoming smartphone APIs that allow low-level exposure control.

The runtime of the algorithm depends mainly on the size of the input images. In fact, we perform $R \times C \times M$ non-blind deconvolutions on patches but as the patches are overlapping, we process somewhat more pixels than the image contains. Our tests were conducted off-line on a PC but each component of our algorithm is portable to a smartphone with little modifications.

7 CONCLUSION

We proposed a new algorithm for unmodified off-the-shelf smartphones for the removal of handshake blur from photographs of planar surfaces such as posters, advertisements, or price tags. We re-formulated the fast non-blind uniform blur removal algorithm of Krishnan and Fergus [Kri09] to multiple input frames and to non-uniform blur. We rendered piecewise uniform blur kernels from the gyroscope measurements and we adaptively weighted the input patches in a multiframe deconvolution framework based on their blurriness. The distortion effects of the rolling shutter of the smartphone camera were compensated prior to deblurring. We applied existing off-line and on-line methods for gyroscope and camera calibration, however, a robust on-line calibration method is still an open question. We have shown the effectiveness of our method in qualitative experiments on images degraded by synthetic and real blur. Our future work will focus on fast blind deblurring that is initialized with our rendered motion blur kernels so that less iterations are required in the traditional multiscale kernel estimation.

ACKNOWLEDGEMENTS

We thank Fabrizio Pece and Tobias Nägeli for the productive discussions and the anonymous reviewers for their comments and suggestions.

8 REFERENCES

[Bae13] H. Bae, C. C. Fowlkes, and P. H. Chou. Accurate motion deblurring using camera motion tracking and scene depth. In *IEEE Workshop on Applications of Computer Vision (WACV)*, 2013.

[Bay08] H. Bay, T. Tuytelaars, and L. van Gool. SURF: Speeded Up Robust Features. In *European Conference on Computer Vision (ECCV)*, 2006.

[Bel14] S. Bell, A. Troccoli, and K. Pulli. A non-linear filter for gyroscope-based video stabilization. In *European Conference on Computer Vision (ECCV)*, 2014.

[3] `https://www.khronos.org/streaminput/`

[Cho09] S. Cho and S. Lee. Fast motion deblurring. In *ACM SIGGRAPH Asia*, 2009.

[Deb97] P. E. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *ACM SIGGRAPH*, 1997.

[For10] P.-E. Forssen and E. Ringaby. Rectifying rolling shutter video from hand-held devices. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[Gol12] A. Goldstein and R. Fattal. Blur-kernel estimation from spectral irregularities. In *European Conference on Computer Vision (ECCV)*, 2012.

[Han11] G. Hanning, N. Forslow, P.-E. Forssen, E. Ringaby, D. Tornqvist, and J. Callmer. Stabilizing cell phone video using inertial measurement sensors. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2011.

[Ito14] A. Ito, A. C. Sankaranarayanan, A. Veeraraghavan, and R. G. Baraniuk. BlurBurst: Removing blur due to camera shake using multiple images. In *ACM Transactions on Graphics*, 2014.

[Jia13] C. Jia and B. Evans. Online calibration and synchronization of cellphone camera and gyroscope. In *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2013.

[Jia14] C. Jia and B. Evans. Online camera-gyroscope autocalibration for cell phones. In *IEEE Transactions on Image Processing*, 23(12), 2014.

[Jos10] N. Joshi, S. B. Kang, C. L. Zitnick, and R. Szeliski. Image deblurring using inertial measurement sensors. In *ACM SIGGRAPH*, 2010.

[Jos08] N. Joshi, R. Szeliski, and D. Kriegman. Psf estimation using sharp edge prediction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[Kar11] A. Karpenko, D. Jacobs, J. Baek, and M. Levoy. Digital video stabilization and rolling shutter correction using gyroscopes. Technical report, Stanford University, 2011.

[Koh12] R. Köhler, M. Hirsch, B. Mohler, B. Schölkopf, and S. Harmeling. Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database. In *European Conference on Computer Vision (ECCV)*, 2012.

[Kri09] D. Krishnan and R. Fergus. Fast image deconvolution using hyper-Laplacian priors. In *Advances in Neural Information Processing Systems (NIPS)*. 2009.

[Lev09] A. Levin, Y. Weiss, F. Durand, and W. Freeman. Understanding and evaluating blind deconvolution algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[Li10] Y. Li, S. B. Kang, N. Joshi, S. Seitz, and D. Huttenlocher. Generating sharp panoramas from motion-blurred videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[Par14] S. Park and M. Levoy. Gyro-based multi-image deconvolution for removing handshake blur. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[Par11] S. H. Park, A. Adams, and E.-V. Talvala. The FCam API for programmable cameras. In *ACM International Conference on Multimedia (MM)*, 2011.

[Rav05] A. Rav-Acha and S. Peleg. Two motion-blurred images are better than one. *Pattern Recognition Letters*, 26(3), 2005.

[Rin14] E. Ringaby and P.-E. Forssen. A virtual tripod for hand-held video stacking on smartphones. In *IEEE International Conference on Computational Photography (ICCP)*, 2014.

[Sin13] O. Sindelar and F. Sroubek. Image deblurring in smartphone devices using built-in inertial measurement sensors. In *Journal of Electronic Imaging*, 22(1), 2013.

[Sin14a] O. Sindelar, F. Sroubek, and P. Milanfar. A smartphone application for removing handshake blur and compensating rolling shutter. In *IEEE Conference on Image Processing (ICIP)*, 2014.

[Sin14b] O. Sindelar, F. Sroubek, and P. Milanfar. Space-variant image deblurring on smartphones using inertial sensors. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2014.

[Sun13] L. Sun, S. Cho, J. Wang, and J. Hays. Edge-based blur kernel estimation using patch priors. In *IEEE International Conference on Computational Photography (ICCP)*, 2013.

[Tai13] Y.-W. Tai, X. Chen, S. Kim, S. J. Kim, F. Li, J. Yang, J. Yu, Y. Matsushita, and M. Brown. Nonlinear camera response functions and image deblurring: Theoretical analysis and practice. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(10), 2013.

[Tai10] Y.-W. Tai, H. Du, M. Brown, and S. Lin. Correction of spatially varying image and video motion blur using a hybrid camera. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6), 2010.

[Wan04] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. In *IEEE Transactions on Image Processing*, 13(4), 2004.

[Why12] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. In *International Journal of Computer Vision*, 98(2), 2012.

[Xio12] Y. Xiong, K. Saenko, T. Darrell, and T. Zickler. From pixels to physics: Probabilistic color derendering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[Xu13] L. Xu, S. Zheng, and J. Jia. Unnatural L0 sparse representation for natural image deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

Multiscopic HDR Image sequence generation

Raissel Ramirez Orozco
Group of Geometry and Graphics
Universitat de Girona (UdG), Spain
rramirez@ima.udg.edu

Céline Loscos
CReSTIC-SIC
Université de Reims Champagne-Ardenne (URCA), France

Ignacio Martin
Group of Geometry and Graphics
Universitat de Girona (UdG), Spain

Alessandro Artusi
Graphics & Imaging Laboratory
Universitat de Girona (UdG), Spain

ABSTRACT

Creating High Dynamic Range (HDR) images of static scenes by combining several Low Dynamic Range (LDR) images is a common procedure nowadays. However, 3D HDR video acquisition hardware barely exist. Limitations in acquisition, processing, and display make it an active, unsolved research topic. This work analyzes the latest advances in 3D HDR imaging and proposes a method to build multiscopic HDR images from LDR multi-exposure images. Our method is based on a patch match algorithm which has been adapted and improved to take advantage of epipolar geometry constraints of stereo images. Up to our knowledge, it is the first time that an approach different than traditional stereo matching is used to obtain accurate matching between the stereo images. Experimental results show accurate registration and HDR generation for each LDR view.

Keywords

High Dynamic Range, Stereoscopic HDR, Stereo Matching, Image Deghosting

1 INTRODUCTION

High Dynamic Range (HDR) imaging is an increasing area of interest at academic and industrial level, and one of its crucial aspects is the reliable and easy content creation with existing digital camera hardware.

Digital cameras with the ability to capture extended dynamic range, are appearing into the consumer market. They either use a sensor capable of capturing an intensity range larger than the one captured by traditional 8-10 bit sensors, or integrate hardware and software improvements to largely increase the acquired intensity range. However, due to their high costs, their use is very limited [BADC11].

Traditional low dynamic range (LDR) camera sensors provide an auto-exposure feature that can be used to increase the dynamic range of light captured from the scene. The main idea is to capture the same scene at different exposure levels, and then to combine them to reconstruct the full dynamic range.

To achieve this, different approaches have been presented [MP95, DM97, RBS99, MN99, RBS03], but they are not exempt of drawbacks. Ghosting effects may appear in the reconstructed HDR image, when the pixels in the source images are not perfectly aligned [TA+14]. This is due to two main reasons: either camera movement or objects movement in the scene. Several solutions for general image alignment exist [ZF03]. However, it is not straightforward to consider such methods because exposures in the image sequence are different, making alignment a difficult problem.

High Dynamic Range content creation is lately moving from the 2D to 3D imaging domain introducing a series of open problems that need to be solved. 3D images are displayed in two main different ways: either from two views for monoscopic displays with glasses or from multiple views for auto-stereoscopic displays. Most of current auto-stereoscopic displays accept from five to nine different views [LLR13]. To our knowledge, HDR auto-stereoscopic displays do not exist yet. We can feed LDR auto-stereoscopic displays with tone-mapped HDR, but we will need at least five different views.

Some of the techniques used for 2D applications have been recently extended for multiscopic images [TKS06, LC09, SMW10, BRR11, BLV+12, OMLA13, OMLA14, BRG+14, SDBRC14]. However, most of these solutions suffer from a common limitation: they need to rely on accurate dense stereo matching between images which may fail in case of different brightness between exposures [BVNL14]. Thus, more robust and faster solutions for matching different exposure images that allow an easy and reliable acquisition of multiscopic HDR content are highly needed.

(a) Non aligned (b) Bätz *et al.* [BRG+14] (c) Our Result

Figure 1: Set of LDR multiview images from the IIS Jumble data-set, courtesy of Bätz [BRG+14]. The top row shows five multiview exposure images, one exposure per view. The bottom row shows HDR images obtained without alignment (a), using Bätz's method (b) and using our proposed patch-match method (c).

In response to this need, we propose in this paper a solution to combine sets of multiscopic LDR images into HDR content using image correspondences based on the Patch Match algorithm [BSFG09]. This algorithm has been used recently by Sen *et al.* [SKY+12] to build HDR images that are free of ghosting effects. The need of improving the coherence of neighbour patches was already presented in [FP10].The results were promising for multi-exposure sequences where the reference image is moderately under exposed or saturated but it fails when the reference image has large under exposed or saturated areas.

We propose to adapt this approach for multiscopic image sequences (Figure 1), that answer to a simplified epipolar geometry obtained by parallel optical axes (images not originally taken with this geometric configuration can be later rectified). In particular, we reduce the search space in the matching process and improving the incoherence problem of the patch-match. Each image in the set of multi-exposed images is used as a reference; we look for matches in all the remaining images. These accurate matches allow to synthesize images corresponding to each view which are merged into one HDR image per view.

Our contributions into the field can be summarized as follows:

- We provide an efficient solution to multiscopic HDR image generation.
- Traditional stereo matching produce several artifacts when directly applied on images with different exposures. We introduce the use of an improved version of patch-match to solve these drawbacks.
- Patch-match algorithm was adapted to take advantage of the epipolar geometry reducing its computational costs while improves its matching coherence drawbacks.

2 RELATED WORK

Two main areas were considered in this work. The following section presents the main state of the art related to stereo HDR acquisition and multi-exposed image alignment for HDR generation.

2.1 Stereo HDR Acquisition

Some prototypes have been proposed to acquire stereo HDR content from multi-exposure views. Most approaches [TKS06, LC09, SMW10, Ruf11, BRG+14, AKCG14] are based on a rig of two cameras placed like a conventional stereo configuration that captures differently exposed images. Troccoli *et al.* [TKS06] propose to use cross correlation stereo matching to get a primary disparity match. The correspondences are used to calculate the camera response function (CRF) to convert pixel values to radiance space. Stereo matching is executed again but now in radiance space to extract the depth maps.

Lin and Chang [LC09] use SIFT descriptors to find correspondences. The best correspondences are selected using epipolar constrains and used to calculate the CRF. The stereo matching algorithm is based on belief propagation to derive the disparity map. A ghost removal technique is used to avoid artifacts due to noise or stereo mismatches. Even though, disparity maps are not accurate in large areas that are under exposed or saturated.

Rüfenacht[Ruf11] compares two different approaches to obtain stereoscopic HDR video content: a temporal

approach, where exposures are captured by temporally changing the exposure time of two synchronized cameras to get two frames of the same exposure per shot, and a spatial approach, where cameras have different exposure times for all shots so that two frames of the same shot are exposed differently.

Bonnard *et al.* [BLV+12] propose a methodology to create content that combines depth (3D) and HDR video for auto-stereoscopic displays. They use reconstructed depth information from epipolar geometry to drive the pixel match procedure. The matching method lacks of robustness especially on under exposed or saturated areas. Akhavan *et al.* [AYG13, AKCG14] offer a useful comparison of the difference between disparity maps obtained from HDR, LDR and tone-mapped images.

Selmanovic *et al.* [SDBRC14] propose to generate Stereo HDR video from a pair HDR-LDR, using an HDR camera and a traditional digital camera. In this case, one HDR view needs to be reconstructed. Three methods are proposed to generate an HDR image: (1) to warp the existing one using a disparity map, (2) to increase the range of the LDR view using an expansion operator and (3) an hybrid of the two methods which provides the best results.

Bätz *et al.* [BRG+14] present a framework with two LDR cameras, the input images are rectified before the disparity estimation. Their stereo matcher is exposure invariant and use Zero-Mean Normalized Cross Correlation (ZNCC) as a matching cost. The matching is performed on the gray-scale radiance space image followed by local optimization and disparities refinement. Some artifacts may persist in the saturated areas.

2.2 Multi-exposed Image Alignment

In the HDR context, most of methods on image alignment focus on movement between images caused by hand-held capture, small movement of tripods or matching moving pixels from dynamic objects in the scene. One of the main drawbacks for HDR video acquisition is the lack of robust algorithms for deghosting. Hadziabdic *et al.* [HTM13], Srikantha *et al.* [SS12] and Tursun *et al.* [TA+14] provide good reviews and comparisons between recent methods.

Kang *et al.* [KUWS03] proposed to capture video sequences alternating long and short exposure times. Adjacent frames are warped and registered to finally generate an HDR frame. Sand and Teller [ST04] combine feature matching and optical flow for spatio-temporal alignment of different exposed videos. They search for frames that best match with the reference frame using locally weighted regression to interpolate and extrapolate image correspondences. This method is robust to changes in exposure and lighting, but it is slow and artifacts may appear if there are objects moving at high speed.

Mangiat and Gibson [MG10] propose to use a method of block-based motion estimation and refine the motion vectors in saturated regions using color similarity in the adjacent frames of an alternating multi-exposed sequence.

Sun *et al.* [SMW10] assume that the disparity map between two rectified images can be modeled as a Markov random field. The matching problem is then posed as a Bayesian labeling problem in which the optimal values are obtained minimizing an energy function. The energy function is composed of a pixel dissimilarity term (using NCC as similarity measure) and a smoothness term which corresponds respectively to the MRF likelihood and the MRF prior.

Sen *et al.* [SKY+12] present a method based on a patch-based energy-minimization formulation that integrates alignment and reconstruction in a joint optimization. This allows to produce an HDR result that is aligned to one of the exposures and contains information from all the rest. Artifacts may appear when there are large under exposed or saturated areas in the reference image.

2.3 Discussion

Stereo matching is a mature research field; very accurate algorithms are available for images taken under the same lighting conditions and exposure. However, most of such algorithms are not accurate for images with important lighting variations. We propose a novel framework inspired by Barnes *et al.* [BSFG09] and Sen *et al.* [SKY+12]. We adapt the matching process to the multiscopic context resulting in a more robust solution.

3 PATCH-BASED MULTISCOPIC HDR GENERATION

Our method takes as input a sequence of LDR images (RAW or not). We transform the input images to radiance space, all the rest of steps are performed using radiance space values instead of RGB pixels. For 8-bits LDR images a CRF per camera needs to be estimated. An overview of our framework is shown in the diagram of the Figure 2. The first step is to recover the correspondences between the **n** images of the set. We propose to use a nearest neighbor search algorithm (see section 3.1) instead of a full stereo matching approach. Each image acts like a reference for the matching process. The output of this step is **n-1** warped images for each exposure. Which then are combined into an output HDR image for each view through a second step (see section 3.2).

3.1 Nearest Neighbor Search

For a pair of images I_r and I_s, we compute a Nearest Neighbor Field (NNF) from I_r to I_s using an improved version of the method presented by Barnes *et*

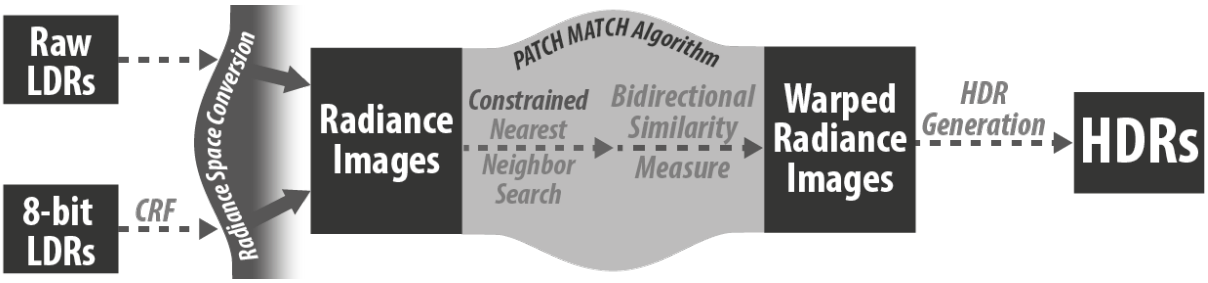


Figure 2: Proposed framework for multiscopic HDR Generation. It is composed by three main steps: (1) radiance space conversion, (2) patch match correspondences search and (3) HDR generation

al. [BSFG09]. NNF is defined over patches around every pixel coordinate in image I_r for a cost function **D** between two patches of images I_r and I_s. Given a patch coordinate $\mathbf{r} \in I_r$ and its corresponding nearest neighbor $\mathbf{s} \in I_s$, $NNF(\mathbf{r}) = \mathbf{s}$. The values of NNF for all coordinates are stored in an array with the same dimensions as I_r.

We start initializing the NNFs using random transformation values within a maximal disparity range on the same epipolar line. Consequently the NNF is improved by minimizing **D** until convergence or a maximum number of iterations is reached. Two candidate sets are used in the search phase as suggested by [BSFG09]: .

(1) *Propagation* uses the known adjacent nearest neighbor patches to improve NNF. It converges fast but it may fall in a local minima.

(2) *Random search* introduces a second set of random candidates that are used to avoid local minima. For each patch centered in pixel v_0, the candidates u_i are sampled at an exponentially decreasing distance from v_i:

$$u_i = v_0 + w\alpha^i R_i \tag{1}$$

where R_i is a uniform random value $\in$ [-1,1], w is the maximum value for disparity search and α is a fixed ratio ($1/2$ is suggested).

Taking advantage of the epipolar geometry both search accuracy and computational performances are improved. Geometrically calibrated images allow to reduce the search space from 2D to 1D domain, consequently reducing the search domain. As an example, using random search we only look for matches in the range of maximal disparity in the same epipolar line (1D domain), avoiding to search in 2D space. This reduces significantly the number of samples to find a valid match.

Typical drawback of the original NNFs approach [BSFG09], used in the patch match algorithm, is the non geometrically coherency of its search results. This problem is illustrated in Figures 3 and 4. Two static neighbor pixels, in the reference image, match two separated pixels in the source image (Figure 3).

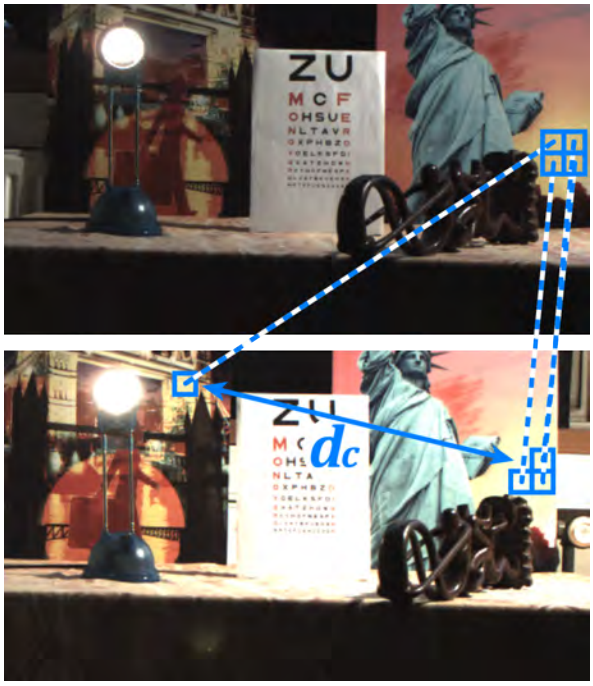


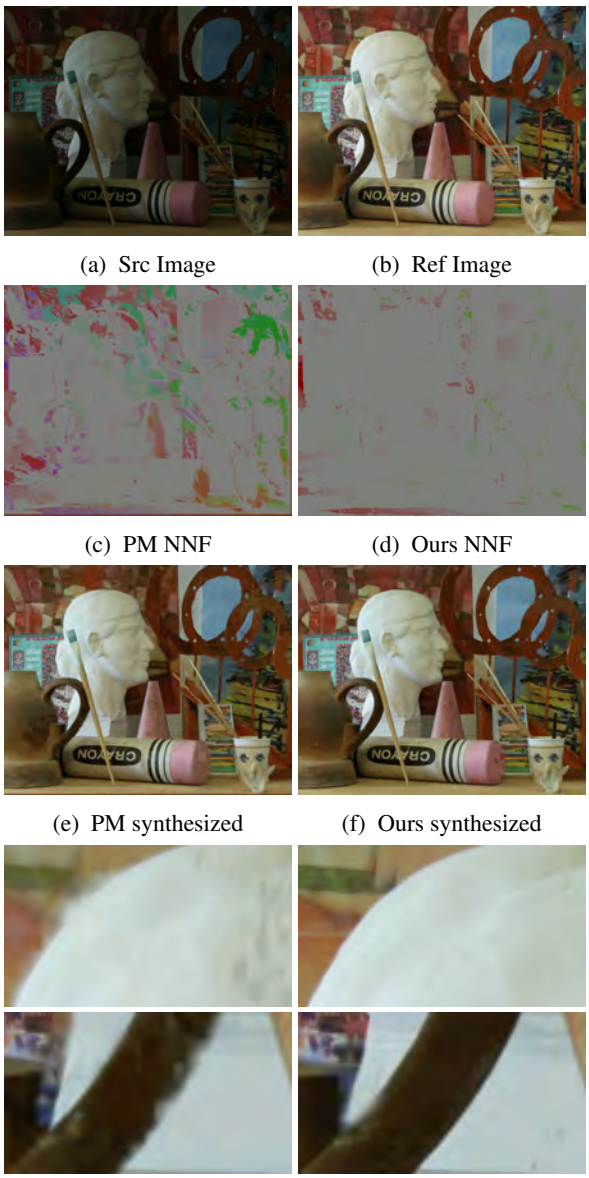
Figure 3: Patches from the reference image (Up) look for their NN in the source image (Down). Even when destination patches are similar in terms of color, matches may be wrong because of geometric coherency problems.

To overcome this drawback we propose a new distance cost function D by incorporating a coherence term to penalize matches that are not coherent with the transformation of their neighbors. Both Barnes *et al.* [BSFG09] and Sen *et al.* [SKY+12] use the Sum of Squared Differences (SSD), described in equation 3 where **T** represents the transformation between patches of **N** pixels in images I_r and I_s. We propose to penalize matches with transformations that differ significantly form it neighbors by adding the coherence term **C** defined in equation 4. The variable d_c represents the Euclidean distance to the closest neighbor's match and Max_{disp} is the maximum disparity value. This new cost function forces pixels to preserve coherent transformations with their neighbors.

$$D = SSD(r,s)/C(r,s) \tag{2}$$

$$SSD = \sum_{n=1}^{N} (I_r - T(I_s))^2 \quad (3)$$

$$C(r,s) = 1 - d_c(r,s)/Max_{disp} \quad (4)$$



(a) Src Image (b) Ref Image

(c) PM NNF (d) Ours NNF

(e) PM synthesized (f) Ours synthesized

(g) Details in (e) (h) Details in (f)

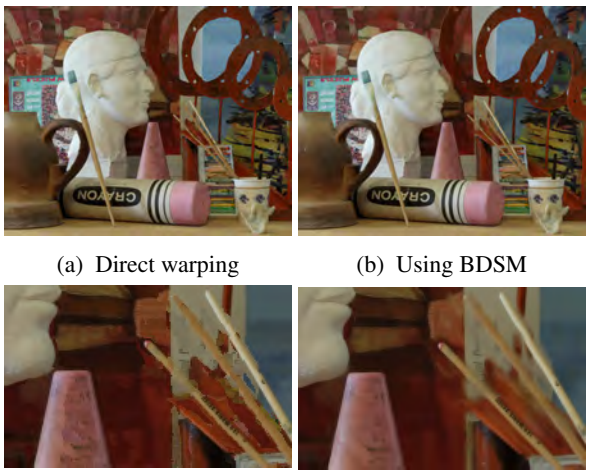
Figure 4: Matching results using original Patch Match [BSFG09] (Left) and our implementation (right) for two iterations using 7x7 patches. Images in the 'Art' dataset courtesy of [vis06]

Figures 4c and 4e show the influence of the coherence problems described in Figure 3 in the matching results. Figures 4d and 4f correspond to the results including the improvements presented in this section. Figures 4c and 4d show a color representation of the NNFs using HSV color space, magnitude of the transformation vector is visualized in the saturation channel and the angle in the hue channel. Areas represented with the same color in the NNF color representation mean similar transformation. Objects in the same depth may have similar transformation. Notice that the original Patch Match [BSFG09] finds very different transformations for neighbor pixels of the same objects and produces artifacts in the synthesized image.

3.2 Warping Images and HDR Generation

The warping images are generated as an average of the patches that contribute to a certain pixel. Direct warping from the NNFs is possible, but it may generate visible artifacts as shown in Figure 5. This is due mainly to incoherent matches between the I_r and I_s images. To solve this problems we use Bidirectional Similarity Measure (BDSM) (Equation 5), proposed by Simakov *et al.* [SCSI08] and used by Barnes *et al.* [BSFG09], which measure similarity between pairs of images. It is defined for every patch $\mathbf{Q} \subset I_r$ and $\mathbf{P} \subset I_s$, and a number $\mathbf{N}$ of patches in each image respectively. It consists of two terms: *coherence* that ensures that the output is geometrically coherent with the reference and *completeness* that ensures that the output image maximizes the amount of information from the source image:

$$d(I_r, I_s) = \overbrace{\frac{1}{N_{I_r}} \sum_{Q \subset I_r} \min_{P \subset I_s} D(Q,P)}^{d_{completeness}} + \overbrace{\frac{1}{N_{I_s}} \sum_{P \subset I_s} \min_{Q \subset I_r} D(P,Q)}^{d_{coherence}} \quad (5)$$



(a) Direct warping (b) Using BDSM

(c) Details in (c) (d) Details in (d)

Figure 5: Images 5a and 5b are both synthesized from the pair in Figure 4. Image 5a was directly warped using values only from the NNF of Figure 4c, which corresponds to matching 4a to 4b. Image 5b was warped using the BDSM of Equation 5 which implies both NNFs of Figures 4c and 4d.

This allows to improve both coherence and consistency by using bidirectional NNFs (from I_r to I_s and backward). It is more accurate to generate images using three iterations in each direction than only six from I_r to

I_s. Using BDSM also prevents artifacts in the occluded areas.

Since the matching is totally independent for pairs of images, it was implemented in parallel. Each image matches all other views. This produces **n-1** NNFs for each view. The NNFs are in fact the two components of the BDSM of equation 5. The new image is the result of accumulating pixel colors of each overlapping neighbor patch and averaging them.

The final HDR image per view is generated using a weighted average [MP95, DM97, MN99] as defined in Equation 6 and the weighting function of Equation 7 proposed by Khan *et al.* [KAR06]:

$$E(i,j) = \frac{\sum_{n=1}^{N} w(I_n(i,j))(\frac{f^{-1}(I_n(i,j))}{\Delta t_n})}{\sum_{n=1}^{N} w(I_n(i,j))} \quad (6)$$

$$w(I_n) = 1 - (2\frac{I_n}{255} - 1)^{12} \quad (7)$$

where I_n represents each image in the sequence, w corresponds to the weight, f is the CRF, Δt_n is the exposure time for the Ith image of the sequence.

4 EXPERIMENTAL RESULTS

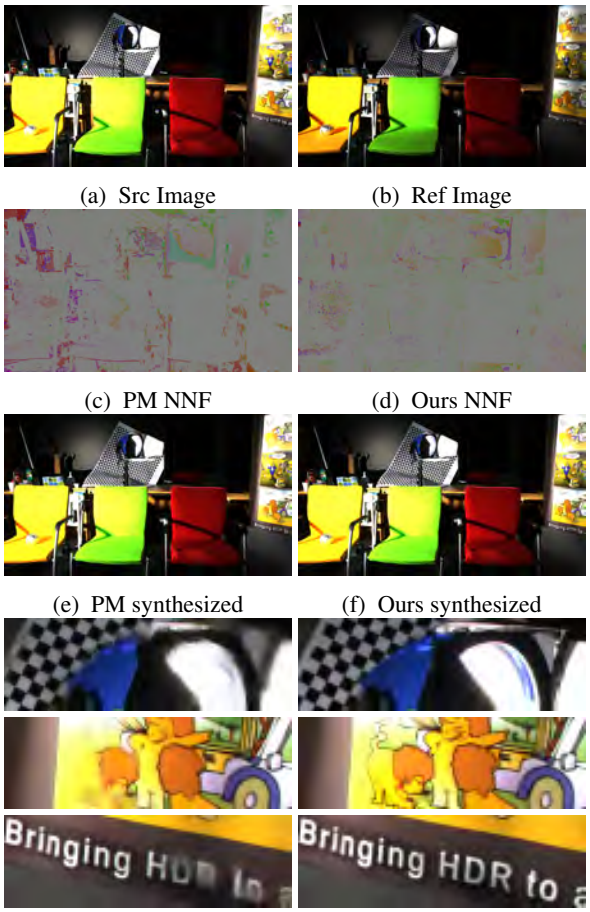
Five data-sets were selected in order to demonstrate the robustness of our results. For the set 'Octo-cam' all the objectives capture the scene at the same time and synchronized shutter speed. For the rest of data-sets the scenes are static. This avoids the ghosting problem due to dynamic objects in the scene. In all figures of this paper we use the different LDR exposures for display purposes only, the actual matching is done in radiance space.

The 'Octo-cam' data-set are eight RAW images with 10-bit of color depth per channel. They were acquired simultaneously using the Octo-cam [PCPD+10] with a resolution of 748x422 pixels. The Octo-cam is a multi-view camera prototype composed by eight objectives horizontally disposed. All images are taken at the same shutter speed (40 ms) but we use three pairs of neutral density filters that reduce the exposure dividing by 2, 4 and 8 respectively. The exposure times for the input sequence are equivalent to 5, 10, 20 and 40 ms respectively [BLV+12]. The objectives are synchronized so all images corresponds to the same time instant.

The sets 'Aloe', 'Art' and 'Dwarves' are from the Middlebury web site [vis06]. We selected images that were acquired under fixed illumination conditions with shutter speed values of 125, 500 and 2000 ms for 'Aloe'and 'Art' and values of 250, 1000 and 4000 ms for 'Dwarves'. They have a resolution of 1390 x 1110 pixels and were taken from three different views. Even if we have only 3 different exposures we can use the seven available views by alternating the exposures like shown in Figure 9.

The last two data-sets were acquired from two of the state of the art papers. Bätz *et al.* [BRG+14] shared their image data set (IIS Jumble) at a resolution of 2560x1920 pixels. We selected five different views from their images. They where acquired at shutter speeds of 5, 30, 61, 122 and 280 ms respectively. Pairs of HDR images like the one in Figure 6, both acquired from a scene and synthetic examples come from Selmanovic *et al.* [SDBRC14]. For 8-bit LDR data sets, the CRF is recovered using a set of multiple exposure of a static scene. All LDR images are also transformed to radiance space for fair comparison with other algorithms.

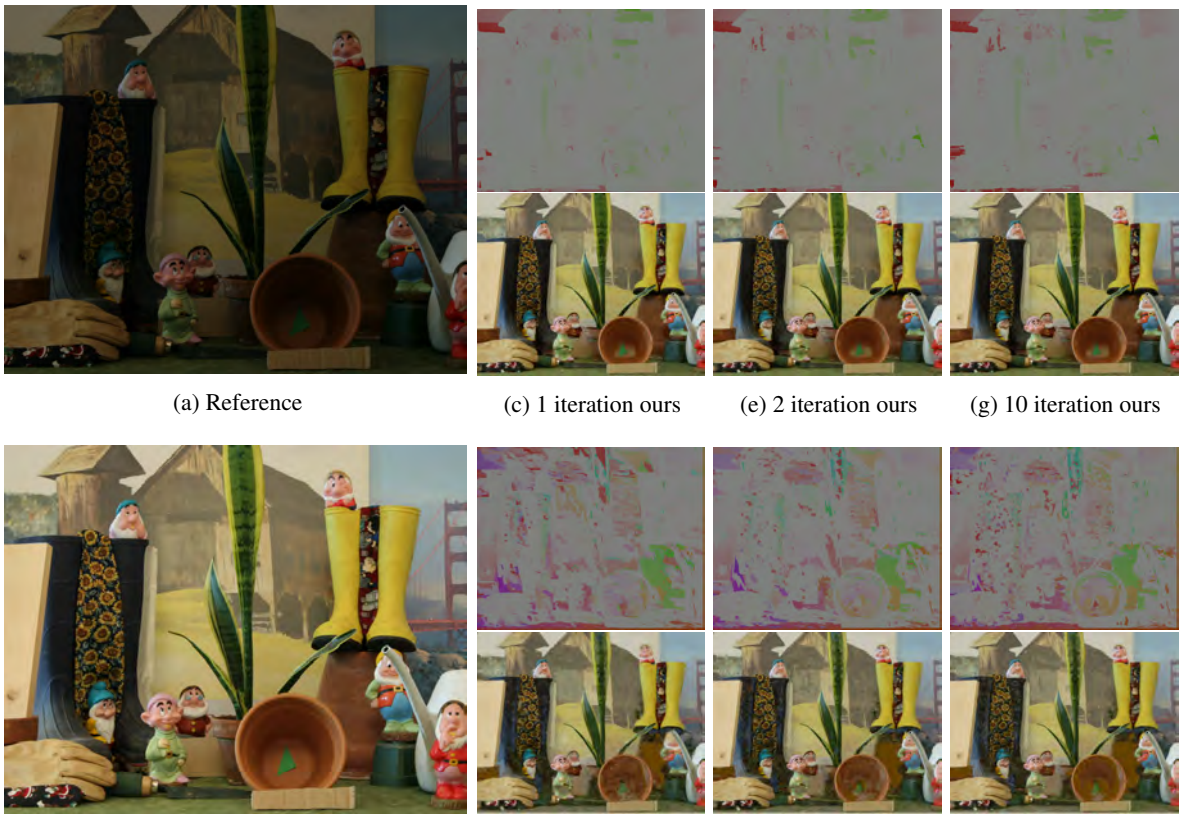
4.1 Results and discussion



(a) Src Image (b) Ref Image (c) PM NNF (d) Ours NNF (e) PM synthesized (f) Ours synthesized (g) Details in (e) (h) Details in (f)

Figure 6: Comparison between original Patch Match and our implementation for two iterations using 7x7 patches. Images 6c and 6d show the improvement on the coherence of the NNF using our method. Images cortesy of [SDBRC14]

Figure 6 shows a pair of images linearized from HDR images courtesy of Selmanovic *et al.* [SDBRC14] and the comparison between the original PM from Barnes *et al.* [BSFG09] and our method including the coherence term and epipolar constrains. The images in Figures 6c and 6d represent the NNF. They are codified into an



(a) Reference (c) 1 iteration ours (e) 2 iteration ours (g) 10 iteration ours

(b) Source (d) 1 iteration PM (f) 2 iteration PM (h) 10 iteration PM

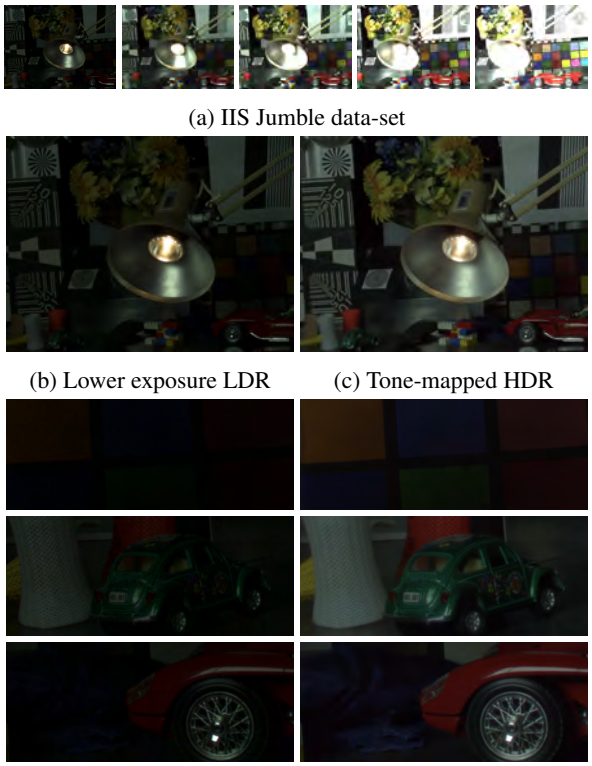
Figure 7: Two images from the 'Dwarves' set of LDR multi-view images form Middlebury [vis06]. Our method with only two iterations achieve very accurate matches. Notice that the original patch match requires more iterations to achieve good results in fine details of the image.

image in HSV color space. Magnitude of the transformation vector is visualized in the saturation channel and the angle in the hue channel. Notice that our result represent more homogeneous transformations, represented in gray color. Images in Figure 6e and 6f are synthesized result images for the **Ref** image obtained using pixels only from the **Src** image. The results correspond to the same number of iterations (2 in this case). Our implementation converges faster producing accurate results in less iterations than the original method.

All the matching and synthesizing process are performed in radiance space. They were converted to LDR using the corresponding exposure times and the CRF for display purposes only. The use of an image synthesis method like the BDSM instead of traditional stereo matching allows us to synthesize values for occluded areas too.

Figure 7 shows the NNFs and the images synthesized for different iterations of both our method and the original patch match. Our method converges faster and produce more coherent results than [BSFG09]. In occluded areas the matches may not be accurate in terms of geometry due to the lack of information. Even in such cases, the result is accurate in terms of color. After several tests, only two iterations of our method were enough to get good results while five iterations were recommended for previous approaches.

Figure 8 shows one example of the generated HDR corresponding to the lowest exposure LDR view in the IIS Jumble data-set. It is the result of merging all synthesized images obtained with the first view as reference. The darker image is also the one that contains more noisy and under-exposed areas. HDR values were recovered even for such areas and no visible artifacts appears. On the contrary, the problem of recovering HDR values for saturated areas in the reference image remains unsolved. When the dynamic range differences are extreme the algorithm does not provide accurate results. Future work must provide new techniques because the lack of information inside saturated areas does not allow patches to find good matches. The CRFs for the LDR images were calculated in a set of aligned multi-exposed images using the software RASCAL, provided by Mitsunaga and Nayar [MN99]. Figure 9 shows the result of our method for a whole set of LDR multi-view and differently exposed images. All obtained images are accurate in terms of contours, no visible artifacts comparing to the LDR were obtained.



(a) IIS Jumble data-set

(b) Lower exposure LDR (c) Tone-mapped HDR

(d) Details in (b) (e) Details in (c)

Figure 8: Details of the generated HDR image corresponding to a dark exposure. Notice that under-exposed areas, traditionally difficult to recover, are successfully generated without visible noise or misaligned artifacts.

Figures 10 show the result of the proposed method in a scene with important lighting variances. The presence of the light spot introduce extreme lighting differences between the different exposures. For bigger exposures the light glows from the spot and saturate pixels not only inside the spot but also around it. There is not information in saturated areas and the matching algorithm does not find good correspondences. The dynamic range is then compromised in such areas and they remain saturated. Our method is not only accurate but faster than previous solutions. [SKY+12] mention that their method takes less than 3 minutes for a sequence of 7 images of 1350x900 pixels. The combination of a reduced search space and the coherence term effectively implies a reduction of the processing time. In a Intel Core i7-2620M 2,70 GHz with 8 GB of memory, our method takes less than 2 minutes (103 $\pm$ 10 seconds) for the Aloe data set with a resolution of 1282x1110 pixels.

5 CONCLUSIONS

This paper presented a framework for auto-stereoscopic 3D HDR content creation that combines sets of multiscopic LDR images into HDR content using image dense correspondences. Methods that, when used for 2D domain cannot be used for 3D HDR content creation without introducing visible artifacts. Our novel approach is extending the well known Patch Match algorithm, introducing an improved random search function that takes advantage of the epipolar geometry. Also a coherence term is used for improving the matching process. These modifications allow to extend the original approach to work for HDR stereo matching, while improving its computational performances. We have presented a series of experimental results showing the robustness of our approach, in the matching process, when compared with the original approach and its qualitative results.

6 ACKNOWLEDGMENTS

The authors would like to thank the reviewers and those who collaborated with us for their suggestions. This work was partially funded by the TIN2013-47137-C2-2-P project from Ministerio de Economía y Competitividad, Spain. Also by the Spanish Ministry of Sciences through the Innovation Sub-Program Ramón y Cajal RYC-2011-09372 and from the Catalan Government 2014 SGR 1232. It was also supported by the EU COST HDRi IC1005 and the SGR 2014 1232 from Generalitat de Catalunya.

REFERENCES

[AKCG14] Tara Akhavan, Christian Kapeller, Ji-Ho Cho, and Margrit Gelautz. Stereo hdr disparity map computation using structured light. In *HDRi2014 Second International Conference and SME Workshop on HDR imaging*, 2014.

[AYG13] Tara Akhavan, Hyunjin Yoo, and Margrit Gelautz. A framework for hdr stereo matching using multi-exposed images. In *Proceedings of HDRi2013 First International Conference and SME Workshop on HDR imaging*, Paper no. 8, Oxford/Malden, 2013. The Eurographics Association and Blackwell Publishing Ltd.

[BADC11] Francesco Banterle, Alessandro Artusi, Kurt Debattista, and Alan Chalmers. *Advanced High Dynamic Range Imaging: Theory and Practice*. AK Peters (CRC Press), Natick, MA, USA, 2011.

[BLV+12] Jennifer Bonnard, Celine Loscos, Gilles Valette, Jean-Michel Nourrit, and Laurent Lucas. High-dynamic range video acquisition with a multiview camera. *Optics, Photonics, and Digital Technologies for Multimedia Applications II*, pages 84360A–84360A–11, 2012.

[BRG+14] Michel Bätz, Thomas Richter, Jens-Uwe Garbas, Anton Papst, Jürgen Seiler, and André Kaup. High dynamic range video reconstruction from a stereo camera setup. *Signal Processing: Image Communication*, 29(2):191 – 202,

Figure 9: Up: 'Aloe' set of LDR multi-view images from Middlebury web page [vis06]. Down: the resulting tone mapped HDR taking each LDR as reference respectively. Notice the coherence between all generated images.

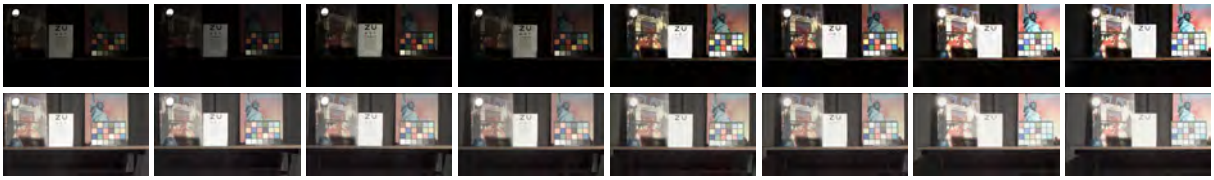


Figure 10: Up: Set of LDR multi-view images acquired using the Octo-cam [PCPD+10]. Down: the resulting tone mapped HDR taking each LDR as reference respectively. Despite the important exposure differences of the LDR sequence, coherent HDR results are obtained. It is important to mention that highly saturated areas remain saturated in the resulting HDR.

2014. Special Issue on Advances in High Dynamic Range Video Research.

[BRR11] Michael Bleyer, Christoph Rhemann, and Carsten Rother. Patchmatch stereo - stereo matching with slanted support windows. In *Proceedings of the British Machine Vision Conference*, pages 14.1–14.11. BMVA Press, 2011. http://dx.doi.org/10.5244/C.25.14.

[BSFG09] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28(3), aug 2009.

[BVNL14] Jennifer Bonnard, Gilles Valette, Jean-Michel Nourrit, and Celine Loscos. Analysis of the Consequences of Data Quality and Calibration on 3D HDR Image Generation. In *European Signal Processing Conference (EUSIPCO)*, Lisbonne, Portugal, 2014.

[DM97] Paul Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *In proceedings of ACM SIGGRAPH (Computer Graphics)*, volume 31, pages 369–378, 1997.

[FP10] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(8):1362–1376, Aug 2010.

[HTM13] Kanita Karaduzovic Hadziabdic, Jasminka Hasic Telalovic, and Rafal Mantiuk. Comparison of deghosting algorithms for multi-exposure high dynamic range imaging. In *Proceedings of the 29th Spring Conference on Computer Graphics*, SCCG '13, pages 021:21–021:28, New York, NY, USA, 2013. ACM.

[KAR06] Erum Arif Khan, Ahmet Oğuz Akyüz, and Erik Reinhard. Ghost removal in high dynamic range images. In *Image Processing, 2006 IEEE International Conference on*, pages 2005–2008, Oct 2006.

[KUWS03] Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High dynamic range video. *ACM Trans. Graph.*, 22(3):319–325, 2003.

[LC09] Huei-Yung Lin and Wei-Zhe Chang. High dynamic range imaging for stereoscopic scene representation. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 4305–4308, 2009.

[LLR13] Laurent Lucas, Celine Loscos, and Yannick Remion. *3D Video from Capture to Diffusion*. Wiley-ISTE, October 2013.

[MG10] Stephen Mangiat and Jerry Gibson. High dynamic range video with ghost removal. *Proc. SPIE*, 7798:779812–779812–8, 2010.

[MN99] Tomoo Mitsunaga and Shree K. Nayar. Radiometric self calibration. *IEEE International Conf. Computer Vision and Pattern Recognition*, 1:374–380, 1999.

[MP95] Steve Mann and R. W. Picard. *On Being undigital With Digital Cameras: Extending Dynamic Range By Combining Differently Exposed Pictures*. Perceptual Computing Section, Media Laboratory, Massachusetts Institute of Technology, 1995.

[OMLA13] Raissel Ramirez Orozco, Ignacio Martin, Celine Loscos, and Alessandro Artusi. Patch-based registration for auto-stereoscopic hdr content creation. In *HDRi2013 - First International*

Conference and SME Workshop on HDR imaging, Oporto Portugal, April 2013.

[OMLA14] Raissel Ramirez Orozco, Ignacio Martin, Celine Loscos, and Alessandro Artusi. Génération de séquences d'images multivues hdr: vers la vidéo hdr. In *27es journées de l'Association française d'informatique graphique et du chapitre français d'Eurographics*, Reims, France, November 2014.

[PCPD+10] Jessica Prévoteau, Sylvia Chalençon-Piotin, Didier Debons, Laurent Lucas, and Yannick Remion. Multi-view shooting geometry for multiscopic rendering with controlled distortion. *International Journal of Digital Multimedia Broadcasting (IJDMB), special issue Advances in 3DTV: Theory and Practice*, 2010:1–11, March 2010.

[RBS99] Mark A. Robertson, Sean Borman, and Robert L Stevenson. Dynamic range improvement through multiple exposures. In *In Proc. of the Int. Conf. on Image Processing (ICIP 1999)*, pages 159–163. IEEE, IEEE, 1999.

[RBS03] Mark A. Robertson, Sean Borman, and Robert L. Stevenson. Estimation-theoretic approach to dynamic range enhancement using multiple exposures. *Journal of Electronic Imaging*, 12(2):219–228, 2003.

[Ruf11] Dominic Rufenacht. *Stereoscopic High Dynamic Range Video*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland, Agost 2011.

[SCSI08] Denis Simakov, Yaron Caspi, Eli Shechtman, and Michal Irani. Summarizing visual data using bidirectional similarity. *IEEE Conference on Computer Vision and Pattern Recognition 2008 (CVPR'08)*, 2008.

[SDBRC14] Elmedin Selmanovic, Kurt Debattista, Thomas Bashford-Rogers, and Alan Chalmers. Enabling stereoscopic high dynamic range video. *Signal Processing: Image Communication*, 29(2):216 – 228, 2014. Special Issue on Advances in High Dynamic Range Video Research.

[SKY+12] Pradeep Sen, Nima Khademi Kalantari, Maziar Yaesoubi, Soheil Darabi, Dan B. Goldman, and Eli Shechtman. Robust patch-based HDR reconstruction of dynamic scenes. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2012)*, 31(6):203:1–203:11, November 2012.

[SMW10] Ning Sun, H. Mansour, and R. Ward. Hdr image construction from multi-exposed stereo ldr images. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, Hong Kong, 2010.

[SS12] Abhilash Srikantha and Désiré Sidibé. Ghost detection and removal for high dynamic range images: Recent advances. *Signal Processing: Image Communication*, page 10.1016/j.image.2012.02.001, February 2012. 23 pages.

[ST04] Peter Sand and Seth Teller. Video matching. *ACM Transactions on Graphics*, 23(3):592–599, August 2004.

[TA+14] Okan Tarhan Tursun, Ahmet Oğuz Akyüz, , Aykut Erdem, and Erkut Erdem. Evaluating deghosting algorithms for hdr images. In *Signal Processing and Communications Applications Conference (SIU), 2014 22nd*, pages 1275–1278, April 2014.

[TKS06] A. Troccoli, Sing Bing Kang, and S. Seitz. Multi-view multi-exposure stereo. In *3D Data Processing, Visualization, and Transmission, Third International Symposium on*, pages 861–868, June 2006.

[vis06] vision.middlebury.edu. Middlebury stereo datasets. `http://vision.middlebury.edu/stereo/data/`, 2006.

[ZF03] Barbara Zitova and Jan Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21:977–1000, 2003.

Occlusion Detection and Index-based Ear Recognition

Madeena Sultana, Padma Polash Paul, Marina Gavrilova

Dept. of Computer Science
University of Calgary
2500 University DR NW
T2N 1N4, Calgary, AB, Canada

{msdeena, pppaul, mgavrilo}@ucalgary.ca

ABSTRACT

Person recognition using ear biometric has received significant interest in recent years due to its highly discriminative nature, permanence over time, non-intrusiveness, and easy acquisition process. However, in a real-world scenario, ear image is often partially or fully occluded by hair, earrings, headphones, scarf, and other objects. Moreover, such occlusions may occur during identification process resulting in a dramatic decline of the recognition performance. Therefore, a reliable ear recognition system should be equipped with an automated detection of the presence of occlusions in order to avoid miss-classifications. In this paper, we proposed an efficient ear recognition approach, which is capable of detecting the presence of occlusions and recognizing partial ear samples by adaptively selecting appropriate features indices. The proposed method has been evaluated on a large publicly available database containing wide variations of real occlusions. The experimental results confirm that the prior detection of occlusion and the novel selection procedure for feature indices significantly improve the biometric system recognition accuracy.

Keywords

Biometric images, occlusion detection, ear recognition, partial occlusion, classifier selection, adaptive feature selection.

1. INTRODUCTION

Biometric authentication offers advantage over traditional PIN (Personal Identification Number) or password-based security since it is harder to forge, steal, transfer, or lose biometric data. At present, biometric based person recognition has enormous demand in government services as well as commercial sectors due to availability of biometric data, enhanced recognition accuracy and non-invasive nature of authentication. Over the last few years, ear biometric has received growing attention and proven to be useful for an automated person recognition [Cha03a], [Che07a]. Unlike face biometrics, ear has no sensitivity to facial expression changes [Kum12a] and it remains almost unchanged throughout the lifetime of a person [Yua12a]. Ear biometric is not only a powerful feature to identify individuals, but also to recognize identical twins [Nej12a]. Moreover, ear has high user acceptance because of its nonintrusive nature and a passive acquisition process [Jai99a]. Similar to other passive biometrics, the recognition performance of ear biometrics may deteriorate significantly due to natural constraints such as occlusion, lightning, pose difference etc. [Bus10a]. Among all natural constraints, occlusion happens to be the most common scenario, since ear is often partially or fully occluded by hair, earrings, headphones, scarfs etc. Information loss due to occlusion is irrevocable. Unlike lightning or pose variations, where some image enhancement techniques can be applied to retrieve partially lost information, the occlusion information loss results in a complete disappearance of a portion of ear. Moreover, distortions of important global features of ear biometrics such as shape and appearance occur, which further undermine the overall system recognition performance. For those reasons, occlusion is one of the most detrimental degrading factors of ear recognition. It has been reported that consideration of un-occluded regions during matching increases recognition accuracy [Yua12a], [Yua12b]. In order to determine the un-occluded portion of ear it is necessary to detect occluded regions. However, detection of occlusions in ear biometrics remained understudied at present. Detection of real occlusions is a very challenging problem since occurrence, locations, proportion, and reasons of occlusion are uncertain. For instance, different regions of an ear may be occluded by different objects such as hair or earrings at the same time. Also, during identification stage, an ear sample may be occluded partially or fully, or may not be occluded at all. In addition, determining the proportion of occlusion is important to make a decision whether the sample is sufficient for recognition process or needs to be reacquired. Last but

not the least, not every method performs equally on different proportions and different regions of occlusions. For example, global features such as shape-based descriptor may perform well in cases of partial occlusion by earrings, while local or block-based features may work better on distorted shapes due to occlusions by hair. Thus, prior detection of the location and proportion of occlusion could help in selecting the appropriate features as well as feature extraction methods. For these reasons, it is important to develop an ear recognition method that is capable of prior detection of occlusion and can select appropriate features for classification at identification stage. In a recent review paper, Pflug and Busch [Pfl12a] pointed out the lack of studies on real-world ear occlusions. This paper fills this niche and provides a solution to this problem by investigating how real occlusion factors such as hair, accessories etc. affects the recognition performance. The novel contributions of this paper are three fold:

1. We propose a novel method for ear occlusion detection and estimation of occlusion degree using skin-color model.
2. We analyze the impact of real ear occlusions (hair and accessories) on recognition performance.
3. We propose a novel index-based partial ear recognition method that utilizes occlusion information adaptively to obtain consistent recognition rate.

The rest of the paper is organized as follows. Section 2 summarizes some existing researches on ear recognitions. The proposed methodology for occlusion detection and ear recognition is described in Section 3. Section 4 demonstrates experimental results of the performance and effectiveness of the proposed method. Finally, concluding remarks and future works are presented in Section 5.

2. RELEVANT WORK

Person identification using ear biometric has drawn significant attention of many researchers over the last decade. Ear biometric has the advantage of a non-intrusive acquisition in a less controlled environment. However, there has always been a tradeoff between the non-invasiveness of image acquisition and its impact on its quality. Restricting the acquisition environment of ear biometric compromises its noninvasive nature and wide acceptance of users. Moreover, noninvasive biometrics are mostly acquired by surveillance cameras, where environment cannot be controlled. Therefore, instead of imposing tight controls on the acquisition environment, the recent research is focused on developing robust biometric systems that can obtain high recognition rates under less than ideal conditions. Occlusion has been studied for face biometrics to some extent [Lin07a], [Taj13a]. However, occlusion conditions, type, area, proportion etc. of ear are very different than face. There is a lack of study on real occlusions of ear biometrics during identification stage. In this section, we will discuss some contemporary ear recognition methods.

In 2010, Bustard and Nixon [Bus10a] proposed a robust method for ear recognition using homographies calculated from the Scale Invariant Feature Transform (SIFT) points. Authors also showed that performance of this method degraded with an increasing proportion of occlusions. However, the method did not include an automated occlusion detection as well as proportion calculation. Experimentation was conducted on simulated occluded conditions and the effect of real-world occlusions remained uninvestigated. Efficient feature extraction of an ear biometric has been investigated in many recent works. For instance, Huang et al. [Hua11a] proposed Uncorrelated Local Fisher Discriminant Analysis (ULFDA) method for ear recognition, which obtained better performance than benchmark Principle Component Analysis (PCA) and Linear Discriminant Analysis (LDA) [Mar01a]. In 2012, Kumar and Wu [Kum12a] proposed an ear recognition method based on gray-level shape features which outperformed Gabor and log-Gabor based methods. Sparse representation of local texture has been proposed by Kumar and Chan [Kum13a] in 2013, which obtained high recognition rates on different databases. However, none of the aforementioned three methods was evaluated under occluded conditions.

Occlusion has been considered by Yuan and Mu [Yua12a], where a local information fusion method was proposed to obtain robustness under partial occlusion. In this work, experimentation was conducted in the simulated occluded condition, i.e. a specific amount of occlusion has been applied artificially to a certain location of the ear images. However, results showed that the recognition performance of this method varied according to the location as well as amount of occlusion. In another work, Yuan et al. [Yua12b] proposed a sparse based method to recognize partially occluded ears. Experimentation was conducted by adding synthetic occluded regions to the original unoccluded images. Results showed that this method obtained 70% recognition rate for 30% occluded regions, whereas performance dropped below 15% with the increase of the occluded portion up to 50%. In another recent work, Morales et al. [Mor13a] showed that performance of SIFT and Dense-SIFT based feature extraction methods also degraded significantly due to the presence of real-world occlusions. In their work, recognition error rate of SIFT and Dense-SIFT features were 2.78% and 2.03%, respectively on IITD Database. However, the corresponding error rates increased to 20.52% and 25.76% under real-world occlusions on West Pomeranian University of Technology Ear Database [WPUTED]. The above

discussion demonstrates that existing ear recognition methods lack the following:

1. Detection of occlusion remained uninvestigated, although recognition rate highly depends on the presence of occlusion.
2. Recognition rate varies with location and proportion of occlusion. There is a lack of study on automated localization and proportion calculation of occlusion.
3. Existing methods are mostly experimented on simulated or synthetically occluded ear samples in predefined locations. The robustness of ear recognition methods need to be evaluated under real occlusions since type, location, and proportion of real-world occlusions might be very different than the simulated cases.

The above points indicate that there is a gap between real-world occlusion detection and occluded ear recognition methods. In this paper, our main goal is to bridge the gap between occlusion detection and occluded ear recognition by proposing a novel ear recognition method that can detect real occlusions and utilize occlusion information adaptively during recognition stage.

3. PROPOSED METHOD

In this paper, we presented an automated approach of occlusion detection, estimation, and un-occluded region extraction. We also proposed a novel index-based ear recognition method, which can efficiently utilize the extracted un-occluded portion of ear. In the real scenarios, enrolled or template images are mostly obtained under human supervision. Therefore, if occlusion occurs, human supervisor can direct the person to reacquire the sample. On the other hand, identification stage is mostly unsupervised and the system process occluded image in case of the absence of automated detection mechanism, which may eventually lead to a false match. This is why we were interested in measuring occlusion during identification stage. A basic flow diagram of the proposed system is shown in Fig. 1. During enrollment index-based features are extracted and stored in feature database along with corresponding indices. During test, occluded and un-occluded portion of the ear are detected automatically. Next, index-based features are extracted from un-occluded portion of test ear sample and similarity is measured with the corresponding features of enrolled images. The final decision has been obtained from the maximum similarity matching score of the test and enrolled samples. Detailed explanation of the proposed method can be found in the following subsections.

3.1 Types of Occlusion

Occlusion in ear images may occur anytime during identification stage due to the presence of hair, scarf/hat, earring, headphones, dust, and so on. Both shape and appearance of ear vary in a very different way based on the type, location, and proportion of occlusions.

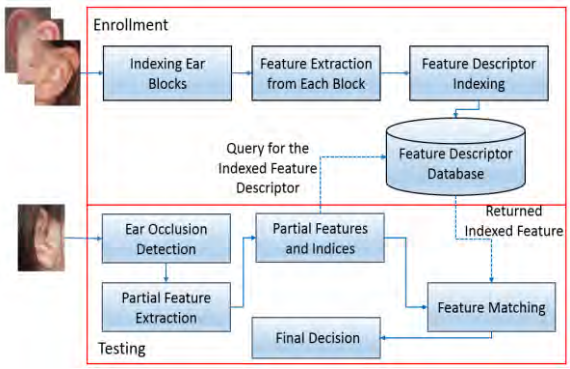


Figure 1. Flow diagram of the proposed ear recognition system.

One reason for the lack of investigation on real occlusions is the unavailability of public database. Researchers [Fre10a] of West Pomeranian University of Technology have created an ear database containing ear samples with different types of real occlusions to facilitate proper validation of ear recognition algorithms. Fig. 2 shows different occluded conditions of ear samples from West Pomeranian University of Technology Ear Database. From Fig. 2, one can see that location, type, and proportion of occlusion are very uncertain and cannot be predefined. Therefore, proper detection of occlusion is indispensable to extract un-occluded features from ear.

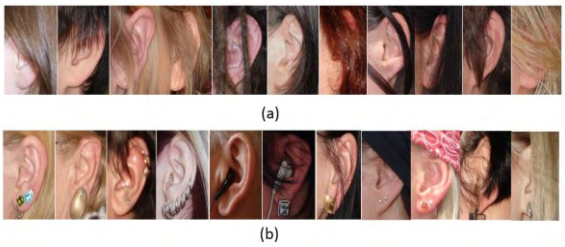


Figure 2. Different types of ear occlusions; a) occlusions by hair; b) occlusions by earrings, scarf, hat, headphones, etc. [WPUTED]

3.2 Ear Enrollment

In this paper, occlusion is considered during test phase to resemble the identification stage. Generally, enrollment is accomplished under human supervision. If occlusion occurs during enrollment human supervisor can reject the biometric sample and reacquire it. Therefore, in this paper, we considered the case that enrolled images are not occluded. Initially, all enrolled images are preprocessed using histogram equalization method and downsampled to 100×80 pixels. Each enrolled image is then partitioned into 10×10 blocks, total 80 blocks. Fig. 3 shows a visual representation of partitioning an ear image into blocks. Next, we applied two-dimensional (2D) Haar

Discrete Wavelet Transform (DWT) to extract local texture features [Sul14a] from each block. Haar wavelet transform decomposes an input block into four sub-bands, one low frequency component (LL) and three detail components (LH, HL, HH). Decomposition to low frequency subband (LL) smoothens image thus reduces noise. Decimated DWT is a popular mathematical tool for image compression since it efficiently reduces image dimensionality at different levels, whereas ensuring seamless reconstructions [DeV92a]. Thus, DWT preserves important information of image while discarding dimensionality. Moreover, DWT is computationally efficient and less sensitive to illumination changes [Sul14a]. The low frequency subband (LL) of DWT contains most of the information of an image. In this work, we applied 1st level Haar DWT to all blocks and considered the low frequency subband of each block as local features. The features of each block are then stored along with its index in feature database.

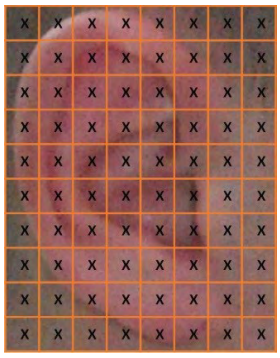


Figure 3. An example of partitioning enrolled ear into indexed-blocks.

3.3 Ear Occlusion Detection

Real-world occlusion detection is a very challenging task because it is uncertain that when and what type of occlusion would arise. There is also no certainty in which portion and what proportion the occlusion would occur. In this section, we propose a novel method of ear occlusion detection and estimation using skin color model. The process in outlined in Algorithm 1.

In our method, the skin color regression model [Pau10a] has been applied for occlusion detection. We utilized skin color model for ear occlusion detection because occlusion obscures skin color information and detection of skin color will allow us to separate occluded and un-occluded regions in ear. The proposed occlusion detection method has four steps: 1) conversion to chromatic color space *r* and *g*, 2) detection of skin regions in *r* and *g* color spaces using skin color likelihood (eq. 5), 3) fusion of *r* and *g* color space images and fill skin regions using morphological operation, and 4) masking un-occluded skin portion from original occluded image. A flow diagram of the steps is depicted in Fig. 5.

Algorithm 1: Occlusion detection and estimation.

Input: Test ear image Y of size $M \times N$.

Output: (B_{Ij}, I_j), un-occluded blocks in Y and corresponding indices.

Step 1: Preprocess Y using histogram equalization method and downsample $M \times N$ to 100×80.

Step 2: Transform image from RGB color space to chromatic color space. Find the value of *r* and *g* as follows [Pau10a]:

$$r = \frac{R}{R+G+B} \quad (1)$$

$$g = \frac{G}{R+G+B} \quad (2)$$

Step 3: Find skin color distribution by 2-D Gaussian model with the following mean vector *A* and covariance matrix *C* [Pau10a]:

$$A = G\{x\}[x = (rg)^T] \quad (3)$$

$$C = \begin{bmatrix} \sigma_{rr} & \sigma_{rg} \\ \sigma_{gr} & \sigma_{gg} \end{bmatrix} \quad (4)$$

Step 4: Estimate likelihood (L) of skin color using the following equation [Pau10a]:

$$L = P(r,g) = \exp[-0.5\,(x-A)^T\, C^{-1}(x-A)\,] \quad (5)$$

where, $x = (r,g)^T$.

Step 5: Find the skin color regions of Y in chromatic color *r* and *g*, denoted as *P(r)* and *P(g)*.

Step 6: Fuse *P(r)* and *P(g)* to obtain resultant binary image, Z = *P(r) AND P(g)*

Step 7: Perform morphological operation using disk shape structuring element of radius 10 to fill the skin regions in Z.

Step 8: Apply Z as a mask on Y to obtain image X containing un-occluded skin portions.

Step 9: Partition X into 10×8 blocks each having 10×10 pixels and construct a block vector {B_i| i= 1, 2, …, 80}.

Step 10: Construct an index vector {I_j| i= 1, 2, …., m}, where B_{Ij} contains skin regions (un-occluded) and m is the total number of un-occluded blocks.

Step 11: Estimate total proportion of occlusion, $E = \frac{\sum_{j=1}^{m} B_{Ij}}{\sum_{i=1}^{80} B_i} \times 100$ (6)

Step 12: If E>60%, discard Y and reacquire test image.

Fig. 6 presents some outcomes of occlusion detection of four ear samples from WPUT Ear Database. Fig. 6 (a) shows four original ear samples containing different types of occlusions due to earring,

headphones, and hair. The corresponding ear samples after chromatic color space conversion are shown in Fig. 6 (b). Fig. 6 (c) presents the resultant skin-regions separated from occlusions. After separating occluded and un-occluded regions, the ear image is partitioned into 80 blocks, each containing 10×10 pixels. The estimated occlusion has been calculated as the ratio of the number of un-occluded blocks over total number of blocks (eq. 6). The estimation of occlusion facilitates auto-rejection of unreliable test images, where most of the information is distorted due to occlusion. In the proposed method, if the estimated occlusion is below 60%, the test image will be used for recognition, otherwise it has to be reacquired. In this way, the proposed ear recognition system can reduce false matches by discarding unreliable test samples, automatically.

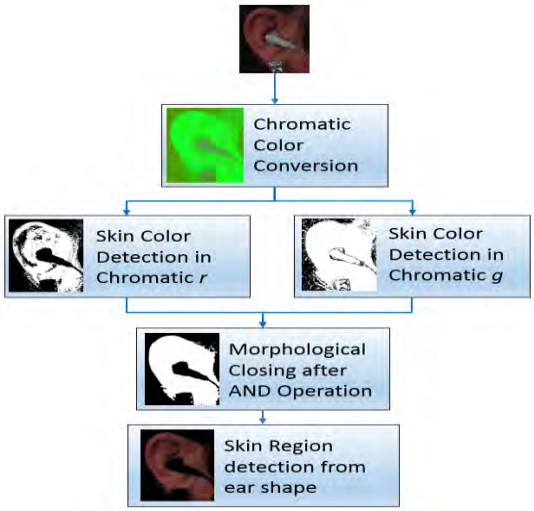


Figure 5. Flow diagram of the four steps of occlusion detection using skin color model.

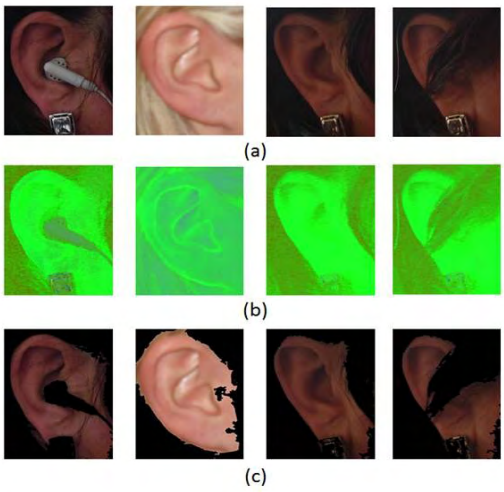


Figure 6. Examples of ear occlusion detection: a) original occluded ears, b) conversion to chromatic color space, c) detected unoccluded skin-regions.

3.4 Partial Feature Extraction and Matching

In the proposed method, partial features are extracted from the detected un-occluded blocks of the test image. 1st level of Haar DWT is applied to all un-occluded blocks (B_{Ij}), and four subbands images (LL, LH, HL, HH) are obtained. The low frequency subband (LL) of each block is considered as the local features of corresponding block and converted to a feature vector.

Finally, similarities between the partial features of test ear and corresponding features of enrolled ears are measured for recognition. Fig. 7 shows an example of the corresponding blocks of a test ear and an enrolled ear. The left image in Fig. 7 shows the blocks of skin regions in test image and the right image shows corresponding blocks in enrolled image. Unlike existing methods, we matched the un-occluded blocks of the detected skin regions to the corresponding blocks of enrolled ears. The index vector (I_j) is used to fetch the corresponding blocks of enrolled ears from feature database. A visual representation of the partial feature extraction and similarity matching process is shown in Fig. 8.

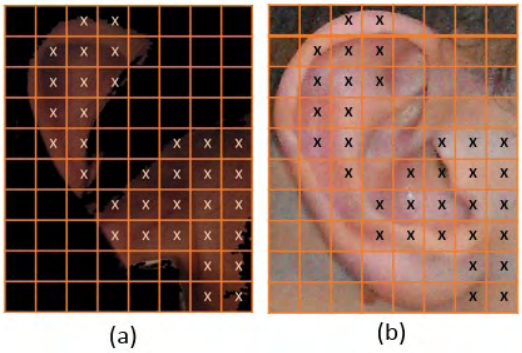


Figure 7. Block indexing, a) unoccluded blocks of test ear, b) corresponding blocks of enrolled ear.

The similarities of the test and enrolled ears are computed using Euclidean distance. Euclidean distance of the indexed features of test ear (V) and enrolled ear (U) can be calculated using the following equation:

$$D = \sqrt{\sum_{j=1}^{m}\sum_{k=1}^{n}\left(u_{jk} - v_{jk}\right)^2} \quad (7)$$

where u_{jk} and v_{jk} are the kth feature of jth block of U and V, respectively, and m is the total number of un-occluded blocks in V. However, a problem may arise during the index-based matching if the indexed blocks of the test ear do not overlap with the indexed blocks of enrolled ear (in other words, if the test ear is shifted to any direction). There are eight possible directions of shift, which is shown in Fig. 9. We propose to solve this problem by using a matching window in all possible eight directions: B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8.

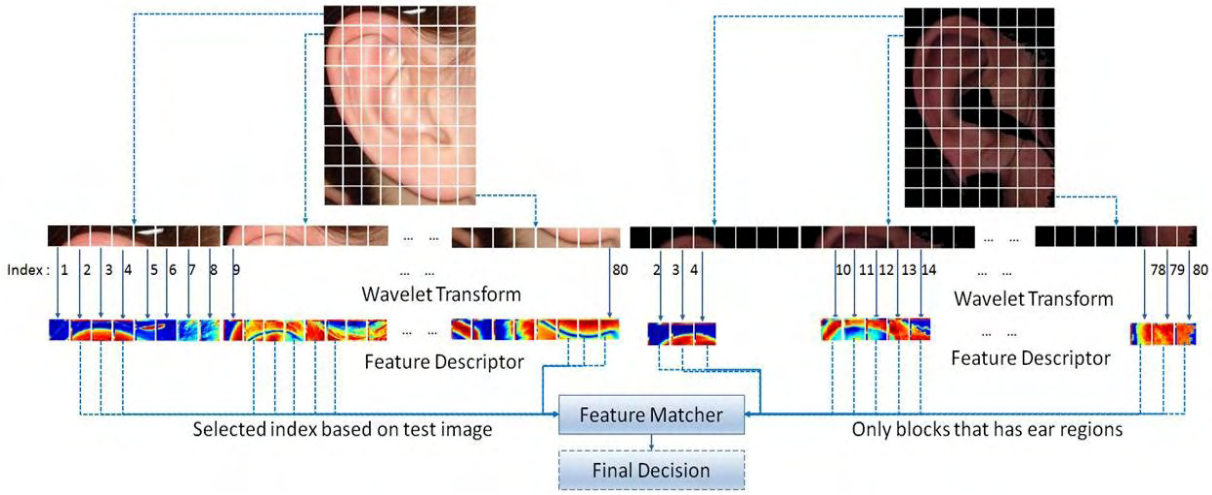


Figure 8. Feature extraction and index-based partial feature matching of test and enrolled ears.

The set of all un-occluded blocks of the test ear is considered as one region. Let us consider $T_{i,j}$ as the un-occluded region of test image and $R_{i,j}$ as the corresponding region in the enrolled sample. The nine similarity score are then calculated using eq. 8 to eq. 16.

$$S_0 = 1 - D(T_{i,j}, R_{i,j}) \quad (8)$$

$$S_1 = 1 - D(T, R_{i,j+1}) \quad (9)$$

$$S_2 = 1 - D(T_{i,j}, R_{i+1,j+1}) \quad (10)$$

$$S_3 = 1 - D(T_{i,j}, R_{i+1,j}) \quad (11)$$

$$S_4 = 1 - D(T_{i,j}, R_{i+1,j-1}) \quad (12)$$

$$S_5 = 1 - D(T_{i,j}, R_{i,j-1}) \quad (13)$$

$$S_6 = 1 - D(T_{i,j}, R_{i-1,j-1}) \quad (14)$$

$$S_7 = 1 - D(T_{i,j}, R_{i-1,j}) \quad (15)$$

$$S_8 = 1 - D(T_{i,j}, R_{i-1,j+1}) \quad (16)$$

The first similarity (S_0) score between the test and enrolled sample is calculated by matching the blocks of test region $T_{i,j}$ and corresponding enrolled region $R_{i,j}$. Next, we calculated the similarity score S_1 along B_1 direction between the test region T_{ij} and training region $R_{i,j+1}$. Then similarity score, S_2 is calculated along B_2 direction between $T_{i,j}$ and $R_{i+1,j+1}$. Similarly, similarity scores S_3 to S_8 are calculated along directions B_3 to B_8 using eq. 11 to eq. 16. The reason for calculating nine similarity scores is that if the test sample is shifted to any of the possible directions, matching score along that direction will be the highest. Thus, calculating similarity scores in all possible directions allow us to find the best matching indices even under shifted condition. Fig. 10 shows pictorial representation of the calculation of nine similarity matching scores from S_0 to S_8. In Fig. 10, S_0 (in middle) represents an example of the corresponding blocks of an enrolled image. The shifted blocks in eights possible directions are represented by S_1 to S_8 in Fig. 10. The shifted blocks were calculated by shifting the whole region (all blocks) towards the eight possible directions as shown in Fig. 9.

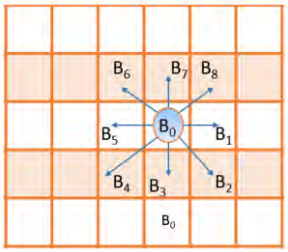


Figure 9. Possible eight directions of image shift.

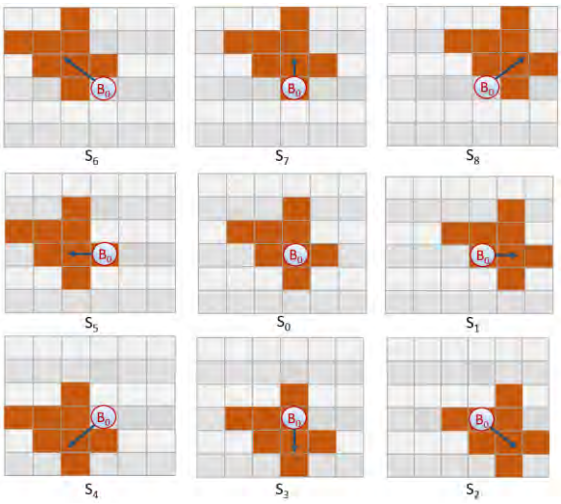


Figure 10. Nine similarity scores (S_0- S_8) calculation by shifting the indexed region of enrolled ear along different directions.

The best matching score is calculated in two ways. First, the highest value among the nine scores is considered as the overall maximum score, S_m (eq. 17). Secondly, we calculated the block-wise maximum score (S_B) among the nine similarity vectors. Calculation of S_B can be shown as eq. 18:

$$S_m = \max_{0 \le i \le 8} S_i \quad (17)$$

$$S_B = \max_{0 \le i \le 8, 1 \le j \le m,} S_{ij} \quad (18)$$

where S_{ij} is the similarity score of jth block of ith similarity vector, m is the maximum number of un-occluded blocks in test image.

4. EXPERIMENTAL RESULTS

Three sets of experiments were conducted to evaluate the performance of the proposed ear occlusion and ear recognition method. All experiments were carried out on Windows 7 operating system, 2.7 GHz Quad-Core Intel Core i7 processor with 16GB RAM. Matlab version R2013a was used for implementation and experimentation of the proposed method. We evaluated our method on WPUT Ear Database [WPUTED] since this is the largest publicly available database containing ear images with wide variations of real occlusions. A brief description of the database is as follows:

WPUT Ear Database [Fre10a]: This database contains 2071 ear images of 254 women and 247 men, total 501 individuals of different ages. There are at least two images per ear of each subject. 15.6% of the images were taken outside and some of them were taken in the dark. 80% of the images are recorded as deformed due to the presence of real occlusions. Ear images of 166 subjects are covered by hair and the presence of earrings are recorded for 147 subjects. The other forms of real-world occlusion in this database are glasses, headdresses, noticeable dirt, dust, birth-marks, ear-pads etc. Many of the samples are simultaneously occluded by different types of occlusion in different proportions.

For our experimentation, the whole database is partition into training and test sets. The training database is created using comparatively un-occluded ear samples. We have single training sample per subject. The occluded images are randomly selected for testing. Each experiment is performed five times and the average recognition accuracy is considered as the recognition performance of the proposed method. Identification rate of the proposed method is analyzed by plotting Cumulative Match Characteristics (CMC) curve. CMC curve is the cumulative probability of obtaining the correct match in the top r positions (ranks). The final matching scores of the test and enrolled images can be obtained in different ways such as block-wise maximum score, overall maximum score, block-wise average score, and overall average score. Therefore, in the first experiment, we compared the performance of the proposed method using different similarity scores to obtain the best performing method of calculating the final similarity score. Fig. 11 shows the CMC curves of the proposed method using block-wise maximum similarity, overall highest similarity, block-wise average score, and overall average similarity scores. From Fig. 11, we can see that the highest performance of the proposed method was obtained by using block-wise maximum similarity score. Consideration of the highest score among the nine scores obtained the 2nd highest performance. Fig. 11 also shows that block-wise average scores performed better than overall average score. However, consideration of the maximum scores are more discriminative than consideration of average scores. The reason for this that not all the similarity scores will find the best match among the test and training blocks and averaging all scores may fade away the best match. Fig. 11 shows that correct matching probabilities of the block-wise maximum similarity, overall maximum similarity, block-wise average similarity, and overall average similarity at rank 1 are 73%, 65%, 57%, and 51%, respectively. Therefore, from the first set of experiments, we found that block-wise maximum score obtained the best results for the proposed method.

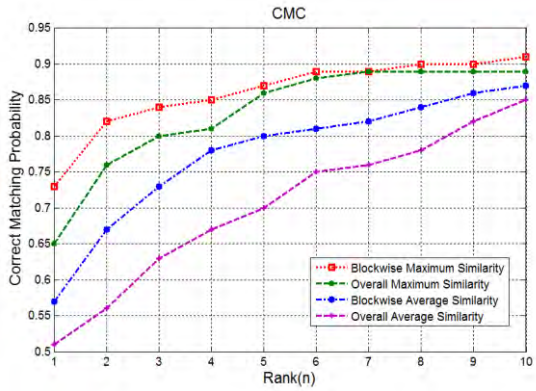


Figure 11. CMC curves of the proposed method using different similarity scores.

In second set of experiments, we compared the performance of the proposed method with a baseline Haar discrete wavelet transform-based method. For the baseline method, ear features were extracted using Haar discrete wavelet transform from test sample without applying any occlusion detection mechanism and the features were matched with the enrolled samples regardless of indices. The CMC curves for the proposed method and the baseline methods are plotted in Fig. 12. From Fig. 12, we can see that the rank 1 recognition rate for the proposed method is 73%, whereas for the baseline method obtained 60% recognition accuracy. Also, 91% recognition rate was obtained by the proposed method within rank 10. The CMC curves in Fig. 12 demonstrate the effectiveness of prior occlusion detection and index-based matching of occluded features.

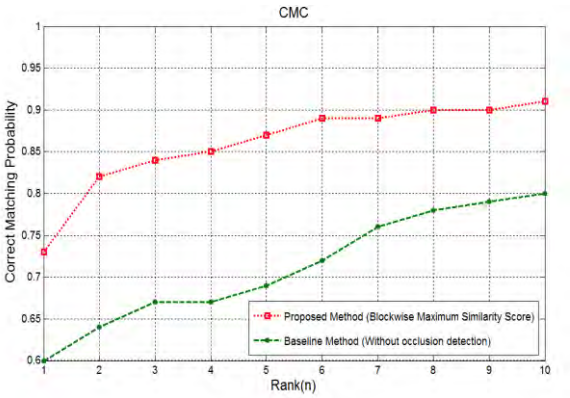


Figure 12. Recognition performance improvement by the proposed method on WPUT database.

In the final set of experiment, we evaluated performance of the proposed method on different amount of occlusions. Amount of occlusion is estimated as the ratio of the occluded blocks over total number of blocks in an ear sample. Fig. 13 shows how the performance of the proposed method varied with different proportion of occlusion. From Fig. 13, we can see that the proposed method can obtain recognition rate as high as 85% with 10% estimated occlusion. Also, the recognition performance remained nearly 80% under 30% occlusion, which is a better result than reported by previous studies. The performance of the proposed method was at 67% even with the 50% of ear image occluded! However, there is simply not enough features for high precision of ear recognition when over 60% of an ear is occluded and in this case ear sample needs to be reacquired.

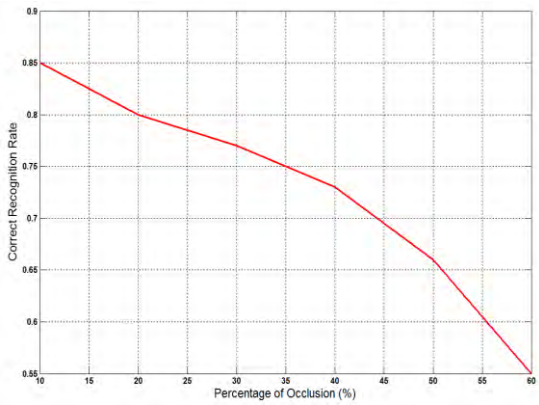


Figure 13. Performance of the proposed method with different degrees of occlusions.

From the above experiments, we can summarize the performance improvement of the proposed method as follows. First of all, automated occlusion detection and estimation allowed us to decide upon whether an ear sample is good enough to be recognized or it is needed to be reacquired. In this way, the proposed method can improve recognition rate by reducing false matches of overly occluded images. Secondly, unlike existing methods where occluded regions were predefined, localization of unoccluded portion in our method is automated. Therefore, the system can adaptively decide upon which portion of the image is unoccluded and good for feature extraction. Thirdly, during recognition, features are extracted from only unoccluded portion of the ear image and matched with corresponding portion of the enrolled samples, which reduces the probability of unreliable matching of the occluded portion. Finally, the problem of shifted indices is solved by using the best block-wise matching scores in eight different scores. For these reasons, the proposed method is capable of obtaining a reliable recognition performance under real occlusions of ears during identification stage.

5. CONCLUSION

A completely automated approach to ear occlusion detection and estimation using skin color model has been proposed in this paper. We also proposed a novel index-based ear recognition method to recognize partially occluded ears effectively. The most important advantage of the proposed method is it can estimate occlusion on ear samples during identification stage and adaptively use this information to select proper indices of the features for recognition purpose. There is a scarcity of occluded ear samples in biometric community and only few publicly available databases contain occluded ear samples. However, the adaptive decision making process of the proposed method doesn't depend on any learning or training of occlusions and thus can be applied to any database. The proposed method of handling ear occlusion was proved to be a very effective in the real world scenarios. Our experiments on real occluded ear images validated the effectiveness of occlusion detection and index-based feature matching for partial ear recognition. Future research will look into incorporating weights into an occlusion estimation process to improve the recognition even further.

6. ACKNOWLEDGMENTS

The authors would like to thank NSERC DISCOVERY program grant RT731064, URGC, NSERC ENGAGE, NSERC Vanier CGS, and Alberta Ingenuity for partial support of this project.

7. REFERENCES

[Bus10a] Bustard, John D., and Nixon, Mark S. Toward unconstrained ear recognition from two-dimensional images. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 40, No. 3, pp. 486-494, 2010.

[Cha03a] Chang, K., Bowyer, K. W., Sarkar, S., & Victor, B. (2003). Comparison and combination of ear and face images in appearance-based biometrics. IEEE Transactions on Pattern Analysis

and Machine Intelligence, 25, No. 9, pp. 1160-1165, 2003.

[Che07a] Chen, H., & Bhanu, B. (2007). Human ear recognition in 3D. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29, No. 4, pp. 718- 737, 2007.

[DeV92a] De Vore, R. A., Jawerth, B., & Lucier, B. J. (1992). Image compression through wavelet transform coding. IEEE Transactions on Information Theory, 38, No. 2, pp. 719-746.

[Fre10a] Frejlichowski, D., and Tyszkiewicz, N. (2010). The West Pomeranian University of Technology ear database–a tool for testing biometric algorithms. In Image analysis and recognition, pp. 227-234, 2010. Springer Berlin Heidelberg.

[Hua11a] Huang, H., Liu, J., Feng, H., & He, T. (2011). Ear recognition based on uncorrelated local Fisher discriminant analysis. Neurocomputing, 74, No. 17, pp. 3103-3113, 2011.

[Jai99a] Jain, Anil K., Bolle, R., and Pankanti, S. eds. Biometrics: personal identification in networked society. Springer Science & Business Media, 1999.

[Kum12a] Kumar, A., and Wu, C., Automated human identification using ear imaging. Pattern Recognition, 45, No. 3, pp. 956-968, 2012.

[Kum13a] Kumar, A., and Chan, T. S. T., Robust ear identification using sparse representation of local texture descriptors. Pattern Recognition, 46, No. 1, pp. 73-85, 2013.

[Lin07a] Lin, D., & Tang, X. Quality-driven face occlusion detection and recovery. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07), pp. 1-7, 2007.

[Mar01a] Martínez, A. M., & Kak, A. C. (2001). PCA versus LDA. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23, No. 2, pp. 228-233, 2001.

[Mor13a] Morales, A., Ferrer, M. A., Diaz-Cabrera, M., & Gonzalez, E. (2013, October). Analysis of local descriptors features and its robustness applied to ear recognition. 47th International Carnahan Conference on Security Technology (ICCST), pp. 1-5, 2013. IEEE.

[Nej12a] Nejati, H., Zhang, L., Sim, T., Martinez-Marroquin, E., & Dong, G. (2012, November). Wonder ears: Identification of identical twins from ear images. 21st International Conference on Pattern Recognition (ICPR), pp. 1201-1204, 2012. IEEE.

[Pau10a] Paul, P. P., Monwar, M. M., Gavrilova, M. L., & Wang, P. S. Rotation invariant multiview face detection using skin color regressive model and support vector regression. International Journal of Pattern Recognition and Artificial Intelligence, 24, No. 08, pp. 1261-1280, 2010.

[Pfl12a] Pflug, A., & Busch, C. (2012). Ear biometrics: a survey of detection, feature extraction and recognition methods. Biometrics, IET, 1, No. 2, pp. 114-129.

[Sul14a] Sultana, M., Gavrilova, M., & Yanushkevich, S. Expression, pose, and illumination invariant face recognition using lower order pseudo Zernike moments. 9th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP), pp. 216-221, 2014.

[Yan03a] Yan, C., Sang, N., & Zhang, T. (2003). Local entropy-based transition region extraction and thresholding. Pattern Recognition Letters, 24, No. 16, pp. 2935-2941.

[Yua12a] Yuan, L., and Chun Mu, Z. Ear recognition based on local information fusion. Pattern Recognition Letters, 33, No. 2, pp. 182-190, 2012.

[Yua12b] Yuan, L., Li, C., and Mu, Z. Ear recognition under partial occlusion based on sparse representation. 2012 International Conference on System Science and Engineering (ICSSE), pp. 349-352, 2012.

[WPUTED] http://ksm.wi.zut.edu.pl/wputedb/ last accessed on March 12, 2015.

[Taj13a] Tajima, Y., Ito, K., Aoki, T., Hosoi, T., Nagashima, S., & Kobayashi, K. (2013, June). Performance improvement of face recognition algorithms using occluded-region detection. 2013 International Conference on Biometrics (ICB), pp. 1-8, 2013.

Performance and Quality Analysis of Convolution-Based Volume Illumination

Paolo Angelelli
University of Bergen
Bergen, Norway
Paolo.Angelelli@UiB.no

Stefan Bruckner
University of Bergen
Bergen, Norway
Stefan.Bruckner@UiB.no

ABSTRACT

Convolution-based techniques for volume rendering are among the fastest in the on-the-fly volumetric illumination category. Such methods, however, are still considerably slower than conventional local illumination techniques.
In this paper we describe how to adapt two commonly used strategies for reducing aliasing artifacts, namely pre-integration and supersampling, to such techniques. These strategies can help reduce the sampling rate of the lighting information (thus the number of convolutions), bringing considerable performance benefits. We present a comparative analysis of their effectiveness in offering performance improvements. We also analyze the (negligible) differences they introduce when comparing their output to the reference method.
These strategies can be highly beneficial in setups where direct volume rendering of continuously streaming data is desired and continuous recomputation of full lighting information is too expensive, or where memory constraints make it preferable not to keep additional precomputed volumetric data in memory. In such situations these strategies make single pass, convolution-based volumetric illumination models viable for a broader range of applications, and this paper provides practical guidelines for using and tuning such strategies to specific use cases.

Keywords

Volume Rendering, Global Illumination, Scientific Visualization, Medical Visualization

1 INTRODUCTION

In recent years different medical imaging technologies, such as computed tomography, ultrasonography and microscopy [5], became capable of generating real-time streams of volumetric data at high frame rates. To visualize such data, volume raycasting [2, 10], capable of displaying surfaces from volumetric data without preprocessing, is often used. This happens in particular in situations where inspection of the acquired data is useful already during the acquisition, such as in 4D Echography where volume rendering of real-time data is employed even for guiding interventions. In these cases conventional direct volume rendering techniques that employ local illumination models are generally used, as they are efficient enough to keep up with the incoming data rate when executed on modern GPU hardware, even when not high end. However, just like in polygonal rendering, rendering volume data using an illumination model that approximates global illumination better than simple local shading models is important for numerous reasons, as recent user studies have demonstrated [11, 17]. Researchers have therefore been very active in the last years in proposing efficient and realistic approximations of global illumination, comprehensively covered in a recent survey by Jonsson et al. [7]. Despite the advances in this field, volumetric illumination methods that offer the best performance rely on expensive preprocessing steps to speed up the rendering by reusing precomputed information. Such preprocessing is not applicable in a number of situations, like, for example, when the volume data to be rendered change continuously, but also when memory constraints (e.g., in the case of portable devices or large datasets) make it preferable not to store an additional precomputed illumination volume.

There is, however, a category of techniques that approximate volumetric lighting (single and sometimes multiple scattering) in the same pass used to generate the image, without the need for preprocessing or storing the whole illumination volume. Nonetheless even the fastest methods in this category are on average six to eight times slower [18, 13] than conventional GPU-based direct volume rendering methods using ray-casting and local illumination models such as Phong shading. This performance penalty can be a serious issue where there are constraints on the computational capacity of the system, or when the rendering pipeline includes additional computationally expensive stages such as volume denoising.

In this paper we focus on convolution-based volumetric illumination models [8, 9, 15, 16, 13], a subcategory of single pass volumetric illumination methods built upon slice-based rendering, that operate by iteratively diffusing the lighting information slice after slice using convolutions. Since the geometry setup using ping-pong buffers is a costly operation and, moreover, the convo-

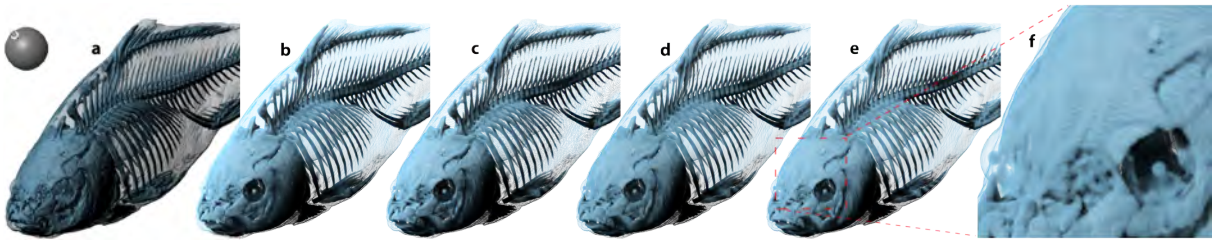


Figure 1: Volume rendering of the carp dataset. **(a)** Raycasting using Phong shading. **(b)** Instant convolution shadows (ICS) with sampling distance of 0.33 voxels (reference). **(c)** ICS with sampling distance of 1 voxel. **(d)** Supersampled convolution shadow (SCS) with a slice distance of 1.3 voxel and 4 tissue subsamples. **(e)** SCS with a slice distance of 1.5 voxels and 5 subsamples (tissue sampling distance of 0.25voxel). **(f)** A closeup highlighting how 1.5 voxels slice distance introduces aliasing artifact despite the dense tissue supersampling. The SCS method, however, allows to increase the inter-slice distance considerably, with an almost linear performance increase. Computation times from left to right: 43ms, 202ms, 88ms, 84ms, 79ms .

lution is performed for every pixel of the view-aligned slices, the sampling distance (and thus the number of slices used for the rendering) and the time necessary for rendering every frame are linearly dependent.

In this paper we analyze the impact of the sampling distance on the performance of this approach in generating aliasing-free images, and incorporate and evaluate the effect of two commonly used strategies to lower this distance: pre-integration [3] and supersampling. The contribution of this paper is therefore twofold. First, we introduce two methods to adapt pre-integration and volume supersampling to convolution-based volumetric illumination techniques, which allow decoupling the sampling rates of the lighting information from the one of the volume. Then we provide a quantitative evaluation of the effects that these strategies have on the performance and practical guidelines for choosing algorithm parameters in order to achieve the best performance without compromising the image quality. We demonstrate that using such strategies can lead to considerable speedups (over 170% in the average case) compared to the standard convolution-based illumination, and, in certain cases, can achieve performance comparable to conventional local illumination methods (see Figure 1 for an example). These performance gains can be instrumental in bringing advanced illumination to volume rendering of streaming data, especially on computationally limited devices, or where the compute unit is used for other computationally expensive steps which are required for the rendering. These strategies can also be beneficial in presence of static data but when, for example, the amount of graphics memory is limited, and precomputing volumetric light information is not preferrable.

2 RELATED WORK

In the area of interactive volume rendering different lighting models to approximate global illumination have been proposed. A thorough overview of such techniques has been provided by Jonsson et al. [7]. In their survey, the authors classify the various techniques in five categories: *local-region-based*, *slice-based*, *light-space-based*, *lattice-based* and *basis-function-based*. Each of these categories describe the underlying paradigm used for calculating volumetric lighting information. The authors also provide a comprehensive analysis of the individual methods, their memory requirements, and their computational costs. The computational costs have been further subdivided into the cost for rendering an image, and the cost for updating the data, the transfer function or the light direction.

For scenarios in which the data is continuously varying we are mostly interested in whether the total time necessary to render the data for the first time exceeds the data rate or not. We therefore adopt a simpler classification here, depending on whether a method requires substantial pre-computation or whether it can produce the final image at interactive frame rates calculating the illumination information on-the-fly. We refer to Jonsson et al. with respect to methods that fit the first of these two classes. In the second class we have splatting-based methods, slice-based methods, and image-plane-sweep-based methods. Splatting was extended to support volumetric lighting by Nulkar and Mueller with the shadow splatting method [12]. This method require an additional pass and the storage of the shadow volume, so it is not an on-the-fly method. However, Zhang and Crawfis [19, 20] later extended the method relaxing these constraints. Still, splatting remains more suitable for sparse or unstructured grids than for dense cartesian grids.

Most of the work in on-the-fly volume illumination can be found in the slice-based category, since synchronization is one of the main issues in calculating the light propagation, and performing slice-based volume rendering implicitly synchronizes the ray front, simplifying the problem. The first method introducing volumetric lighting using this rendering paradigm was *half-angle slicing*, presented by Kniss et al. [8, 9]. The key

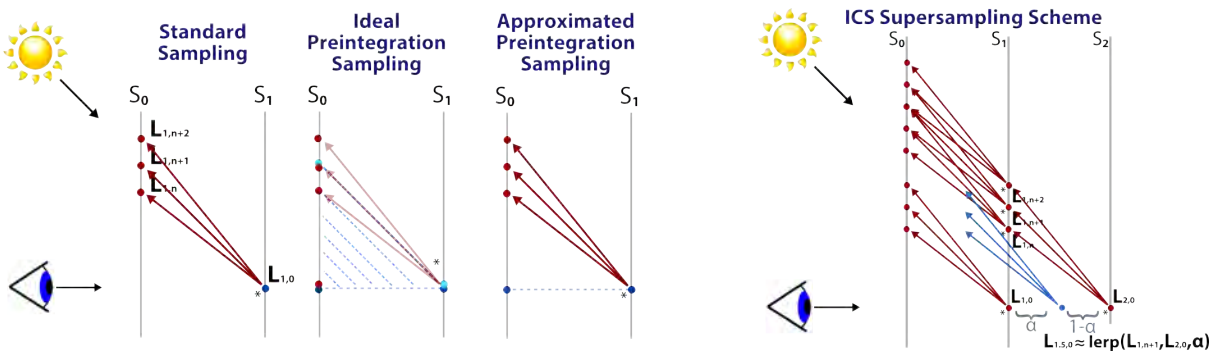


Figure 2: Illustration of our modified schemes for pre-integration (left) and supersampling (right). Correct pre-integration should include in the lookup table also the entry and exit light value. We propose to approximate it performing only tissue pre-integration and sampling the light at S1 position. We also propose to use linearly interpolated light values from the two previous light buffers to calculate the illumination for supersampled tissue samples.

concepts of this method were the implementation of a backward-peaked phase function by iterative convolution and the selection of the slicing direction half-way (hence the name) between viewing and light direction. Schott et al. [15] later presented the *directional occlusion shading* method, constrained to headlight setups to use view-aligned slices, and the same technique to implement the backward-peaked phase function via iterative convolution. However, unlike half-angle slicing, directional occlusion shading does not need two rendering passes per slice. This method was later extended by Šoltészová et al. [16], to allow variable light directions while keeping view-aligned slices. The authors called it *multidirectional occlusion shading*, and also illustrated the advantages of using view-aligned slices in terms of image quality as opposed to half-angle slicing. This method was further improved by Patel et al. [13] with their *instant convolution shadows* method, by using an optimized convolution kernel and allowing the integration of polygonal geometry, making it suitable for volumetric detail mapping to geometrical models.

In the last category, the first and currently only method presented was by Sunden et al. [18], with the *image plane sweep* volume illumination technique. In this method ray-casting is chosen over slice-based rendering, and the rays are not traversed simultaneously, but serialized in a sweep over the image plane. The sweep direction is dependent on the light direction so that the ray direction is orthogonal to the light and subsequent rays can make use of light contributions from previous rays. In their paper the authors show that the performance of their method is similar to half-angle slicing. In this work we focus on slice-based iterative convolution methods.

The last aspect to discuss is how to analyze the results of volume rendering techniques. One of the goals that we have in this work is to improve performance while maintaining the generated images free of aliasing. We identify the optimal parameter setting for the different sampling distances (that is, the most efficient setting that yield aliasing-free images) in a qualitative manner. However, quantitative theoretical models to evaluate the amount of error in volume rendering due to discretization also exist, like the one proposed by Etiene et al. [4], or the method to determine proper sampling frequency of function compositions proposed by Bergner et al. [1]. Performance-wise it has been a common practice to compare different methods on the same viewport size, sampling distance and transfer function, averaging the rendering times over several frames from different viewing direction [14, 18]. In this work we adopt the same strategy. *Timings are averaged over several frames and the viewport size is always fixed to 512x512 pixels*.

3 METHOD

To explain how to adapt supersampling and pre-integration for a convolution-based volumetric illumination model, we can use the Instant Convolution Shadow (ICS) method [13] as the reference model. The basic idea of ICS is that each volume sample on a slice acts as light occluder but also as shadow receiver. This means that every sample which, after classification, maps to a non-fully transparent color, will cast shadows onto the next slice. To compute the amount of light that is transmitted from slice n to a position on slice $n+1$, the incoming light on slice n is first attenuated by the opacity of the samples on slice n, and then this outgoing light is convolved with a kernel $k(x)$. This operation is iterated for every pixel on every slice, and the iterative process propagates the lighting information to the end of the scene.

3.1 Pre-integrated ICS

Pre-integration [3] works by assuming linear variation between two consecutive volume samples. It is then

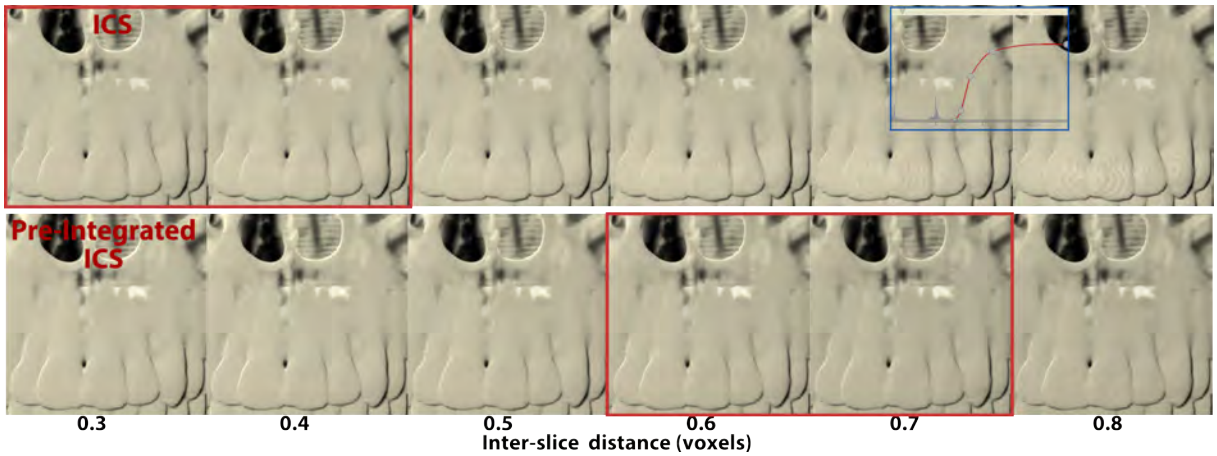


Figure 3: Assessment of the largest sampling distance to produce aliasing-free images for one scene. The transitions where noticeable aliasing appeared are shown in red. Using pre-integration produced identical images and, as expected, allowed to significantly increase the sampling distance while still preventing aliasing.

possible to precompute the volume rendering integral between all possible combination of data values, and store it in a 2D lookup table. During rendering, a simple 2D texture lookup is used. In practice, this approach enables to use of much higher sample distances without noticeable artifacts [3]. However, the basic pre-integration method does not consider illumination, as the resulting increase in dimensionality of the lookup table would make the approach impractical. Previous work [6] showed how to combine local gradient shading with pre-integration by combining two 2D look-up tables. In case of non-local volumetric lighting this is not possible, as the light information depends on the neighborhood of a fragment (see Figure 2).

For this reason we suggest to use standard pre-integration and ignore lighting in the pre-computation. This requires only the conventional 2D lookup table. In this approximation the light propagation proceeds as in the conventional ICS, but the opacity used to attenuate the light comes from the pre-integrated value. We analyze the effect that this approximation has on the image quality, and to what extent it allows us to reduce the inter-slice distance in Section 5.

3.2 Supersampled ICS

The second strategy to increase the distance between slices (and hence, the number of convolutions performed), while still sampling the volumetric function at a sufficiently high rate is to acquire additional volume samples between consecutive slices. The rationale behind this approach is that the color and opacity contributions between consecutive slices are still taken into account, but the illumination propagation is performed at a lower frequency. Such a strategy has pros and cons as compared to pre-integration, where the color is calculated using a finer integration step, but on approximated scalar field values, varying linearly between the front and the back sample. However, these two strategies can also be combined. In order to adapt supersampling to a slice-based renderer with convolution-based lighting, it is necessary to define what light contribution these additional samples collected in between two subsequent slices should receive. The correct solution is illustrated in In Figure 2 on the right (blue convolution). Since this convolution is not possible to calculate due to missing data, we propose an approximation scheme for the light contribution on the additional samples by using their position α in between the slices (see Figure). We then linearly interpolate the light contribution of the current and previous light using this position as the weight.

4 TECHNICAL REALIZATION

Both of these strategies have been shown to be effective in reducing aliasing artifacts, indirectly allowing larger sampling distances. In the specific case of volumetric lighting by convolution shadows, our proposed adaptations blend in the algorithm and are compatible with additional features such as variable light direction, multiple light sources (which can greatly benefit from lower sampling distances), non-white lights or chromatic shadows.

To quantify the benefits that pre-integration and supersampling can provide, we integrated them into a reference implementation of the ICS method. We chose this method because it introduces a number of optimizations over similar methods previously published [16, 15], both from a performance and from an image quality point of view, as discussed in Section 2. The ICS method can be therefore considered one of the most efficient convolution-based volumetric shadows techniques available at the moment.

The necessary adaptations consists of two main ingredients: a loop in the fragment shader to collect the ad-

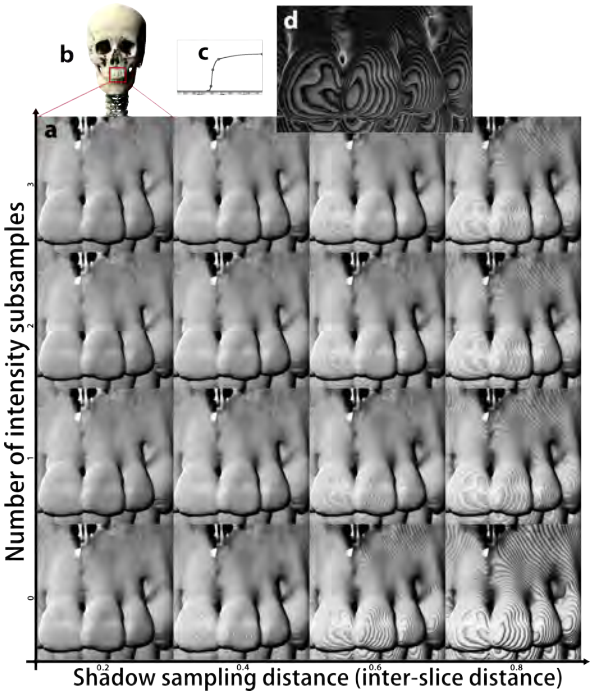


Figure 4: **(a)** Evaluation of the 2D parameter space for supersampled ICS. On the x-axis the inter-slice distance and on the y-axis the number of subsamples are shown. Due to the integral number of possible subsamples, we use x-increments of 0.2 voxels to keep the volume sampling distance identical on the diagonal. Note how increasing the number of substeps does not prevent aliasing anymore after exceeding a certain slice distance. Note that the zoomed views have been desaturated and auto-leveled to enhance the aliasing artifacts, making them easier to see in print. **(b)** Rendering of the whole dataset. **(c)** The transfer function used to geneate these images (same as in Figure 3. **(d)** Absolute differences between the bottom left and the bottom right view (multiplied by a factor of 10 for better visibility). Quantitative measurements are given in Table 1.

ditional samples and an additional color attachment to carry ahead the value of 2 light buffers. However, it should be noted that, if we discard refraction effects that change the color of the light when it propagates in the media, the additional color attachment is not necessary as the light attenuation, even for non-white light, could be approximately described by a single scalar value.

5 RESULTS

5.1 Analysis setup

We carried out a thorough analysis of the different ICS compositing strategies in order to obtain quantitative performance results. To analyze the speedup that these strategies have, we used the average frame rendering time over 100 frames from different view points for different illumination techniques. We compared conventional ICS, ICS with supersampling only for the volume, which from now on will be referred to as Supersampled Convolution Shadows (SCS), pre-integrated ICS and pre-integrated SCS. As a baseline, we also included a conventional volume ray caster with and without local illumination (Phong shading) in the comparison. We conducted our experiments using five different dataset/transfer function combinations. These were a CT dataset of a carp (see Figure 1), a CT dataset of a human head, used with two different transfer functions, one to reveal the skin and one to reveal the skeleton, a CT dataset of a human abdomen revealing the skeleton and the vessels due to contrast agent, and finally a cardiac ultrasound dataset. The dimensions of these volumes are given in Figure 5. The goal of this analysis was to evaluate the performance of each of these techniques in producing artifact-free images. We ran the tests on a workstation equipped with an Intel Core2Quad 2.5GHz CPU, 12GB of RAM and an nVidia Quadro K5000 GPU with 4GB of VRAM. The size of the viewport was fixed to 512x512 pixels.

# Samples / Distance	0	1	2	3
0.2	569ms	572ms	575ms	578ms
0.4	311ms	312ms	314ms	315ms
0.6	225ms	225ms	226ms	227ms
0.8	180ms	181ms	181ms	182ms
0.2	0.0	0.0033	0.0041	0.0045
0.4	0.0060	0.0030	0.0034	0.0035
0.6	0.0118	0.0065	0.0055	0.0054
0.8	0.0187	0.0102	0.0089	0.0083

Table 1: Performance and error analysis for Fig.4. The first table illustrates the necessary time to generate a frame. The second table shows the average pixel difference between the image in the bottom left corner and every other. Pixels have normalized values in the [0,1] interval.

5.2 Parameter Space

We designed the analysis as a two-stage process. In the first analysis stage we estabilished the largest sampling distance for the intensity volume that would still produce aliasing-free pictures using the raycaster, the ICS renderer and the pre-integrated ICS renderer, and used this parameter later on as reference in the performance measurements. This distance was not always the same for the raycasting technique and the ICS technique (slice-based), as these two methods exhibit different aliasing patterns. In particular, and as expected, pre-integrated ICS could consistently tolerate a larger inter-slice distance, which provide an advantage over standard ICS in terms of performance (see Figure 5). This distance was also dependent on the dataset and the transfer function used, so we defined it separately

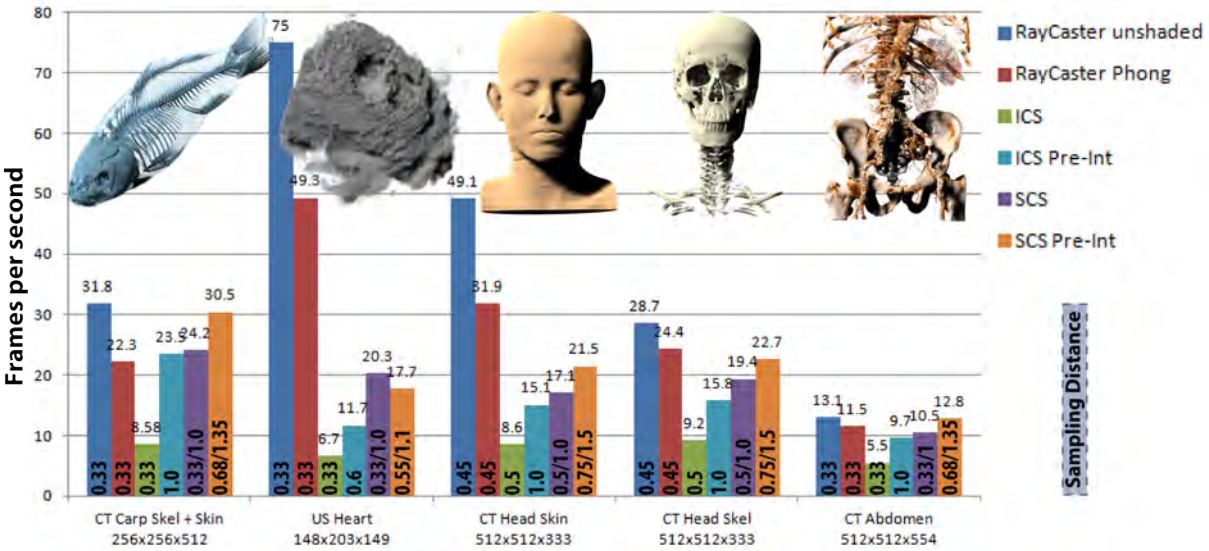


Figure 5: Performance comparison between different rendering methods for five different scenes, depicted on top of each group. On the bottom the size of the volumes in voxels.

for each scene. Figure 3 exemplifies this step for one of the five analyzed scene, in which we qualitatively assessed the larger inter-slice distance that would provide aliasing-free results (for methods to quantitatively assess the amount of aliasing in a rendered image see Section 2).

After the baseline inter-slice distance was identified, we generated the reference images for each of the scenes. In the second step of the analysis we explored the 2D parameter space for the SCS method, in which one dimension is the the inter-slice distance (or the volumetric illumination sampling distance), and the other is the volume sampling distance. However, since our method for integrating supersampling into convolution-based techniques is not able to freely decouple these two parameters (we can only use an integer number of equidistant subsamples between two consecutive sampling slices), we decided to use the number of subsamples as the second parameter in this space. The volume sampling distance can be determined using the formula $SampleDistance = \frac{SliceDistance}{n.o.Subsamples+1}$. Figure 4 shows the result of this exploration for one particular scene using non-preintegrated SCS. This stage was meant to identify the setting of these two parameters that would enable the generation of images identical to the reference most efficiently. After this second stage, optimal parameters for the raycaster, ICS, SCS, pre-integrated ICS and pre-integrated SCS were available, and the performance measurement described in Section 5.1 were conducted using the determined values.

5.3 Analysis results

Figure 5 illustrates the performance that each technique is able to achieve in producing aliasing-free images. When comparing to standard ICS, these results show

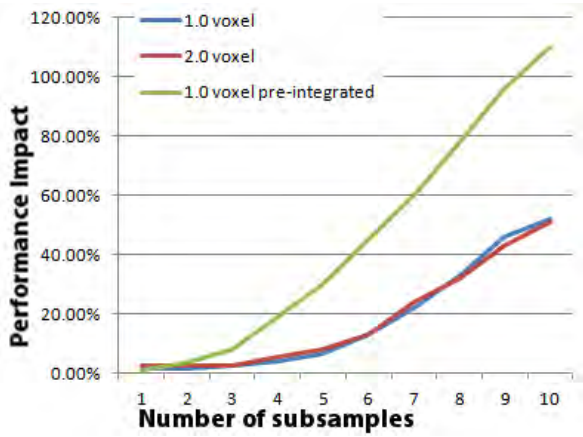


Figure 6: Chart of the performance impact with increasing number of subsamples. In our experiments the slice distance did not play a role, but using pre-integration caused the performance to drop much faster, while regular supersampling comes almost for free for up to 3-4 subsamples.

an average performance increase of 137% for SCS. The worst case scenario for the SCS method has been the CT abdominal scene, where it could offer only a 90% speed increase. In other scenes, in particular in presence of sharper transfer functions such as with the carp dataset or the cardiac ultrasound dataset, the performance increase exceeded 200%.

When using pre-integration, the performance increase over standard ICS is slightly lower despite the usage of same inter-slice distance as SCS in most cases, and even the gathering of only one additional sample as compared to standard ICS (versus the two or three of the SCS method). This behavior can be explained by the fact that sampling a 2D pre-integration table is

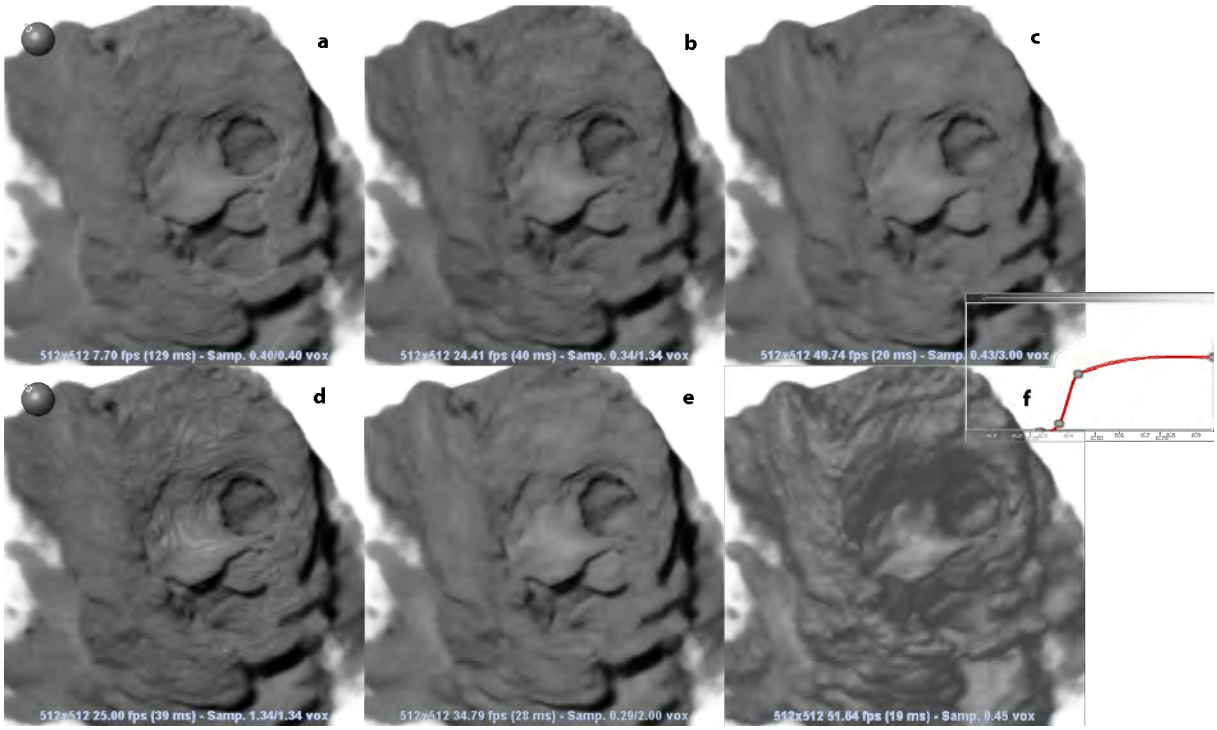


Figure 7: Effect of supersampling a cardiac ultrasound dataset. **(a,d)** ICS with different slice distances. **(b,e,c)** SCS with 1.34 , 2 and 3 voxel slice distances. **(f)** Phong shading for comparison. Note how shadow details on the surfaces progressively disappear with increased sampling distances while shadows casted far away remain the same.

more costly, as the plot in Figure 6, which graphs the penalty for each additional sample for both SCS and pre-integrated SCS, also shows.

Finally, when using both pre-integration and supersampling we could increase the inter-slice distance further without causing aliasing or getting noticeable artifacts in the shading. This combination almost always provided the best performance, except for the cardiac ultrasound dataset, where the inter-slice distance for pre-integration could not be increased as much as in the other scenes. From this analysis we could conclude that, in the average case, the volumetric lighting sampling frequency can be at least halved, when compared to tissue sampling frequency. This possibly due to the lower frequency of the illumination function compared to the post-classified volumetric data. Furthermore we also noticed that the ratio of shadow sampling distance / tissue sampling distance can be further increased in presence of sharper transfer functions.

6 DISCUSSION AND CONCLUSION

Convolution shadow methods and other single pass volumetric illumination techniques can be the only viable option to enable volumetric illumination in a number of application scenarios like real-time 4D echography. Such methods are however constrained on the volume sampling rate by the distance between consecutive slices, requiring a high number of slices for transfer functions containing high frequencies, which consumes a large amount of off-chip GPU memory bandwidth, impacting negatively on the performance. In this work we showed that, by decoupling the sampling rate of the volume from the one of the illumination, we can exploit the fact that illumination is typically less sensitive to lower sampling rates.

We adapted and analyzed two techniques, pre-integration and supersampling, to lower the inter-slice distance and, with some constraints, decouple the two sampling rates. We showed how decoupling these two sampling rates allows less frequent costly convolution operations, bringing a substantial performance increase.

We also discovered that the performance increase using this strategy grows with steeper transfer functions. Both of the strategies analyzed in this paper proved effective, and the most interesting aspect is that, except for one case, they work better when combined. We also experienced that, in certain situations (see Figure 7 for an example), lowering the inter-slice distace beyond what produces images identical to the reference does not immediately introduce aliasing, but the quality of the shading decreases and differences become noticeable. This could however be an acceptable compromise in some situations, in exchange of an additional performance gain.

7 REFERENCES

[1] Steven Bergner, Torsten Moller, Daniel Weiskopf, and David J Muraki. A spectral analysis of function composition and its implications for sampling in direct volume visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 12(5):1353–1360, 2006.

[2] Robert A Drebin, Loren Carpenter, and Pat Hanrahan. Volume rendering. In *ACM Siggraph Computer Graphics*, volume 22, pages 65–74. ACM, 1988.

[3] Klaus Engel, Martin Kraus, and Thomas Ertl. High-quality pre-integrated volume rendering using hardware-accelerated pixel shading. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware*, pages 9–16. ACM, 2001.

[4] Tiago Etiene, Daniel Jonsson, Timo Ropinski, Carlos Scheidegger, Joao Comba, L Nonato, R Kirby, Anders Ynnerman, and C Silva. Verifying volume rendering using discretization error analysis. *Visualization and Computer Graphics, IEEE Transactions on*, 20(1):140–154, Jan 2014.

[5] Anthony W. P. Fitzpatrick, Sang Tae Park, and Ahmed H. Zewail. Exceptional rigidity and biomechanics of amyloid revealed by 4d electron microscopy. *Proceedings of the National Academy of Sciences*, 110(27):10976–10981, 2013.

[6] Amel Guetat, Alexandre Ancel, Stephane Marchesin, and J-M Dischler. Pre-integrated volume rendering with non-linear gradient interpolation. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):1487–1494, 2010.

[7] Daniel Jönsson, Erik Sundén, Anders Ynnerman, and Timo Ropinski. A survey of volumetric illumination techniques for interactive volume rendering. In *Computer Graphics Forum*. Wiley Online Library, 2013.

[8] Joe Kniss, Simon Premoze, Charles Hansen, and David Ebert. Interactive translucent volume rendering and procedural modeling. In *IEEE Visualization 2002*, pages 109–116. IEEE, 2002.

[9] Joe Kniss, Simon Premoze, Charles Hansen, Peter Shirley, and Allen McPherson. A model for volume lighting and modeling. *Visualization and Computer Graphics, IEEE Transactions on*, 9(2):150–162, 2003.

[10] Marc Levoy. Display of surfaces from volume data. *Computer Graphics and Applications, IEEE*, 8(3):29–37, 1988.

[11] Florian Lindemann and Timo Ropinski. About the influence of illumination models on image comprehension in direct volume rendering. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):1922–1931, 2011.

[12] Manjushree Nulkar and Klaus Mueller. Splatting with shadows. In *Proceedings of the 2001 Eurographics conference on Volume Graphics*, pages 35–50. Eurographics Association, 2001.

[13] Daniel Patel, Veronika Šoltészová, Jan Martin Nordbotten, and Stefan Bruckner. Instant convolution shadows for volumetric detail mapping. *ACM Transactions on Graphics*, 32(5):154:1–154:18, 2013.

[14] Timo Ropinski, C Doring, and Christof Rezk-Salama. Interactive volumetric lighting simulating scattering and shadowing. In *Pacific Visualization Symposium (PacificVis), 2010 IEEE*, pages 169–176. IEEE, 2010.

[15] Mathias Schott, Vincent Pegoraro, Charles Hansen, Kévin Boulanger, and Kadi Bouatouch. A directional occlusion shading model for interactive direct volume rendering. In *Computer Graphics Forum*, volume 28, pages 855–862, 2009.

[16] Veronika Šoltészová, Daniel Patel, Stefan Bruckner, and Ivan Viola. A multidirectional occlusion shading model for direct volume rendering. *Computer Graphics Forum*, 29(3):883–891, 2010.

[17] Veronika Šoltészová, Daniel Patel, and Ivan Viola. Chromatic shadows for improved perception. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Non-Photorealistic Animation and Rendering*, pages 105–116. ACM, 2011.

[18] Erik Sundén, Anders Ynnerman, and Timo Ropinski. Image plane sweep volume illumination. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2125–2134, 2011.

[19] Caixia Zhang and Roger Crawfis. Volumetric shadows using splatting. In *Visualization, 2002. VIS 2002. IEEE*, pages 85–92. IEEE, 2002.

[20] Caixia Zhang and Roger Crawfis. Shadows and soft shadows with participating media using splatting. *Visualization and Computer Graphics, IEEE Transactions on*, 9(2):139–149, 2003.

Modeling of Predictive Human Movement Coordination Patterns for Applications in Computer Graphics

Albert Mukovskiy
Section for Computational Sensomotorics, Department of Cognitive Neurology, Hertie Institute for Clinical Brain Research & Centre for Integrative Neuroscience, University Clinic, Tübingen, Germany
albert.mukovskiy@medizin.uni-tuebingen.de

William M. Land
Department of Neurocognition and Action, Bielefeld University, Germany. College of Education and Human Development, The University of Texas at San Antonio,One UTSA Circle, San Antonio, TX 78249, USA.
william.land@utsa.edu

Thomas Schack
CITEC Center of Excellence "Cognitive Interaction Technology", CoR Lab Research Institute of Cognition and Robotics, Department of Neurocognition and Action, Bielefeld University, Germany
thomas.schack@uni-bielefeld.de

Martin A. Giese
Section for Computational Sensomotorics, Department of Cognitive Neurology, Hertie Institute for Clinical Brain Research & Centre for Integrative Neuroscience, University Clinic, Tübingen, Germany
martin.giese@uni-tuebingen.de

Abstract

The planning of human body movements is highly predictive. Within a sequence of actions, the anticipation of a final task goal modulates the individual actions within the overall pattern of motion. An example is a sequence of steps, which is coordinated with the grasping of an object at the end of the step sequence. Opposed to this property of natural human movements, real-time animation systems in computer graphics often model complex activities by a sequential concatenation of individual pre-stored movements, where only the movement before accomplishing the goal is adapted. We present a learning-based technique that models the highly adaptive predictive movement coordination in humans, illustrated for the example of the coordination of walking and reaching. The proposed system for the real-time synthesis of human movements models complex activities by a sequential concatenation of movements, which are approximated by the superposition of kinematic primitives that have been learned from trajectory data by anechoic demixing, using a step-wise regression approach. The kinematic primitives are then approximated by stable solutions of nonlinear dynamical systems (dynamic primitives) that can be embedded in control architectures. We present a control architecture that generates highly adaptive predictive full-body movements for reaching while walking with highly human-like appearance. We demonstrate that the generated behavior is highly robust, even in presence of strong perturbations that require the insertion of additional steps online in order to accomplish the desired task.

Keywords

computer animation, movement primitives, motor coordination, action sequences, prediction.

1 INTRODUCTION

A central problem in computer animation is the online-synthesis of complex behaviors that consist of sequences of individual actions, which have to adapt to continuously changing environmental constraints. An example is the online planning of coordinated walking and reaching, when the position of the reaching goal is dynamically changing.

A prominent approach for the solution of this problem in computer graphics is the adaptive interpolation between motion-captured example actions [WP95, GSKJ03, AFO03]. Other approaches are based on learned low-dimensional parameterizations of whole body motion, which are embedded in mathematical frameworks for the online generation of motion (e.g. [HPP05, SHP04, RCB98, WFH08, LWS02]). Several methods have been proposed that segment action streams into individual actions, where models for the individual actions are adapted online in order to fulfill additional constraints, such obstacle avoidance or the correct positioning of end-effectors ([KGP02, RGBC96, PSS02]). The dependencies between constraints in such action sequences have been recently exploited to generate more realistic animations. In [FXS12] captured motion examples are blended according to a prioritized "stack of controllers". In [SMKB14] the instantaneous blending weights of controllers are pre-specified differently for different body parts involved in the current action and

the priority of the different controllers is governed by their sequential order. In [HK14] the synthesis of locomotion plus arm pointing at the last step is carried out by blending of captured actions determining the weights by "inverse blending optimization". In this study arm pointing was blended with the arm swinging motion of the last step. The choice of the the arm pointing primitives depended on the gait phase, according to an empirical rule introduced by authors.

Physics-based animation is another approach for the online generation of motion (e.g. [ST05, FP03]). Complex action sequences are segmented into individual actions, which are characterized by solutions of optimization problems, derived from mechanics and additional constraints (contact, friction, or specified via-points) ([AMJ07, LHP05, MLPP09]). While these approaches generate highly adaptive behavior for individual actions, the problem to generate natural-looking transitions between the individual actions is non-trivial. As consequence, artifacts (e.g. hesitation, jerky movement) can emerge at transition points, (e.g. [WZ10]).

Opposed to these approaches skilled human motor behavior has been shown to be highly predictive. Within complex activities, action goals and the associated constraints influence actions that appear already a long time before the constraint within the behavioral stream, and thus allows the generation of smooth and optimized behaviors over complex action sequences. This was investigated, for example, in a recent study on the coordination of walking and reaching. Human subjects had to walk towards a drawer and to grasp an object, which was located at different positions in the drawer. Humans optimized their behavior already significantly before object contact, consistent with the hypothesis of maximum end-state comfort during the reaching action [WS10, Ros08], and steps prior to the reaching were modulated in order to accomplish the goal.

Whole body movements of humans and animals are organized in terms of muscle synergies or movement primitives [Ber67, FH05]. Such primitives characterize the coordinated involvement of subsets of the available degrees of freedom in different actions. An example is the coordination of periodic and non-periodic components of the full-body movements during reaching while walking, where behavioral studies reveal a mutual coupling between these components [CG13, CMCH96, Ros08, MB01]. The realism and human-likeness of synthesized movements in robotics and computer graphics can be improved by taking such biological constraints into account [FMJ02].

We present a learning-based framework that makes some of these properties applicable for realtime animation in computer graphics. The underlying architecture is simple and approximates complex full-body movements by dynamic movement primitives that are formulated in terms of nonlinear dynamical systems [GMP$^+$09, PMSG09]. These primitives are constructed from kinematic primitives, that are learned from trajectory sets by anechoic demixing in an unsupervised manner. Similar to the related approaches in robotics [GRIL08, BRI06], the method generates complex movements by the combination of a small number of learned dynamical movement primitives [OG11, GMP$^+$09]. We demonstrate this approach by the highly adaptive online generation of multi-step sequences with coordinated arm movements.

The paper is structured as follows: After the description of the animation system in section 2, we present some example results section 3, followed by a conclusion.

2 SYSTEM ARCHITECTURE

Our work is based on motion capture data from a single human subject performing a drawer opening task. In the following, this data set is described briefly. Then the different key elements of the proposed algorithm are introduced: movement generation by dynamic primitives, modeling of coordination by step-wise regression, and the algorithms for online blending and control.

2.1 Motion capture data

Our system was based on motion capture data from a single human subject that executed a drawer opening task, walking towards a drawer and then reaching for an object in the drawer. The distance of the subject from the drawer and the position of the object was varied [LRSS13] (Fig. 1). These training sequences consisted of three subsequent actions or movements: 1) a normal walking step; 2) a shortened step with the left-hand starting to reach towards the drawer. This step showed a high degree of adaptability, and was typically adjusted in order to create an optimum distance from the drawer (maximum comfort) for the reaching movement during the last action; 3) the drawer opening and the reaching of the object while standing. The object position in the drawer was indicated to the participants at the beginning of each trial. (See [LRSS13] for further details). (See video [**Demo**[1]].)

The analysis of the distances between the pelvis and the drawer or the object in these action sequences reveals the predictive nature of human movement planning, as shown in Fig. 2 where the distances ordered according to the initial walking distance to the drawer. While the length of the first step and the distance from the drawer in the last step are relatively constant, a major distance adjustment is made in the second step.

[1] www.uni-tuebingen.de/uni/knv/arl/avi/wscg15/v1.avi

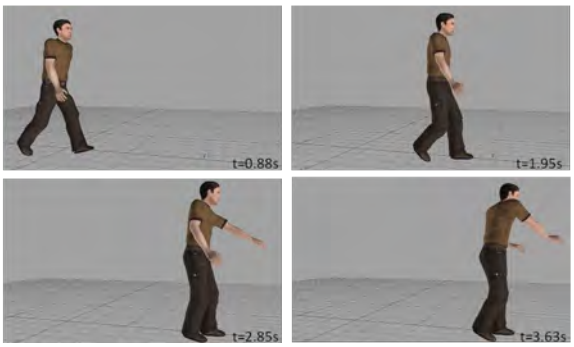


Figure 1: Illustration of the human behavior. The figure illustrates important intermediate postures (normal walking step, step with initiation of reaching, standing while drawer opening, and object reaching).

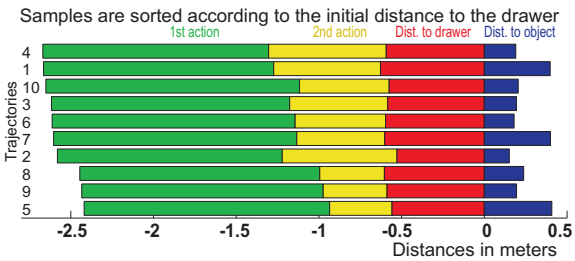


Figure 2: Predictive planning in real human trajectories. Distances from the pelvis to the front panel of the drawer (green, yellow, red), and the distance between the front panel and the object (blue) for different trials. Mainly the second action is adjusted as function of the initial distance from the goal.

The length of the first step is not significantly correlated with the initial distance to the drawer (linear regression: $R^2 = 0.08, p = 0.429$), while the correlations with the distance to the drawer after first step, and the length of the second step are highly significant ($R^2 = 0.95, p = 1.4 \cdot 10^{-6}$).

2.2 Real-time synthesis of movements by learned dynamic primitives

The modeling of the individual actions within the sequence exploits a learning-based approach, which we implemented successfully before for locomotion as well as to other complex human body movements [GMP+09]. The system architecture is illustrated in Fig. 3.

Based on the motion capture data, we learned spatio-temporal components of the three actions in an unsupervised way, applying anechoic demixing [OG11, CdEG13]). We have shown before that this method leads to highly compact approximations of human trajectories, reaching almost perfect approximations of often with less than five learned source functions. The skeleton model of the animated characters had 17 joints. The joint angle trajectories were represented by normalized quaternions (exploiting an exponential map representation, c.f. [Mai90], with 3 variables specifying each

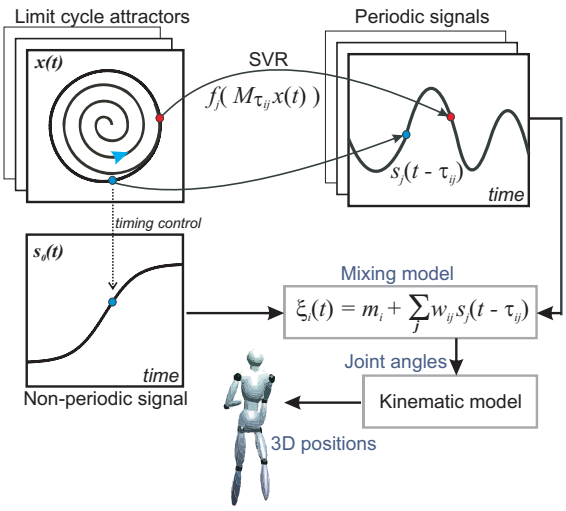


Figure 3: Architecture for the online synthesis of body movements using dynamic primitives.

quaternion). The angles were approximated by an anechoic mixture model of the form:

$$\underbrace{\xi_i(t)}_{\text{angles}} = m_i + \sum_j w_{ij} \underbrace{s_j(t - \tau_{ij})}_{\text{sources}} \tag{1}$$

The index i specifies the joint-angle component, and the index j the source signals s_j. The parameters w_{ij} and τ_{ij} specify the mixing weights and time delays of the source decomposition model, which are estimated together with the other parameters by the demixing algorithm. The parameters m_i specify the means of the joint trajectories.

In order to generate movements online, the source functions are generated by mapping the solutions of a nonlinear dynamical system (canonical dynamics) onto the source functions s_j. For mathematical convenience, we chose a limit cycle oscillator (Hopf oscillator) as canonical dynamics. It can be characterized by the differential equation system (with ω defining the eigenfrequency), for the pair of state variables $[x(t), y(t)]$:

$$\dot{x}(t) = [1 - (x^2(t) + y^2(t))]x(t) - \omega y(t) + k(x_p(t) - x(t))$$
$$\dot{y}(t) = [1 - (x^2(t) + y^2(t))]y(t) + \omega x(t) + k(y_p(t) - y(t))$$

The last terms specify coupling terms to a pair of input signals $x_p(t)$ and $y_p(t)$, and k is the coupling strength. For $k = 0$ this equation produces a stable limit cycle. The state space variables x and y are mapped onto the source functions s_j by nonlinear mapping functions $f_j(x, y)$, which were learned by support vector regression (using a radial basis function kernel and the LIBSVM Matlab® library [CL01]). The learned source functions $s_j(t)$ and corresponding states $[x(t), y(t)]$ from the attractor solution of the limit cycle oscillator were used as training data.

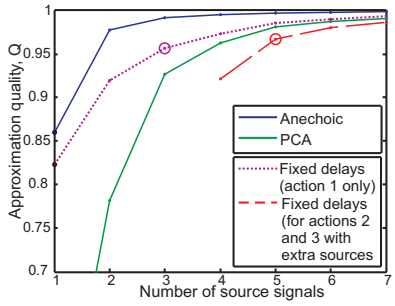


Figure 4: Comparison of approximation quality for different methods for blind source separation as function of the number of sources, using a step-wise regression approach (residuals after subtraction of the contribution of the non-periodic source signal). *Solid lines*: Approximation quality for trajectories of all three actions as a function of the number of (periodic) source functions for anechoic demixing (*blue*) and principle component analysis (PCA) (*green*). The *purple* dotted line shows the approximation quality for the first action, fixing the delays across trials. The *red* dashed line shows approximation quality when 2 additional sources (with fixed delays) were included in order to model the remaining residuals. Circles mark the chosen numbers of sources in our implementation.

The coupling term (for $k > 0$) allows the coupling of different dynamic primitives, if they are specified by the state variables of another oscillator. We have discussed elsewhere that this form of coupling, with appropriate constraints for the parameters, allows to guarantee the stability of the solutions of networks of such primitives. The relevant stability conditions can be derived using *Contraction theory* [LS98, PMSG09].

In our architecture we used one leading oscillator, and the other oscillators were coupled to this leading oscillator in the described form (star topology of the coupling graph, where couplings are unilateral from the center to the leaves of the star). The stability properties of this form of coupling were studied in detail in [PMSG09], and it can be shown that this dynamics has only a single exponentially stable solution. The state of the leading oscillator was also used for the control of the non-periodic source functions.

From the source signals that were generated online, the joint angles were computed using equation (1). Exploiting the fact that the attractor solution of the Hopf oscillator lies on a circle in state space, the delays can be replaced by an appropriate rotations of the variables of the state space (x, y). In this way, we obtained a dynamics without explicit time delays, avoiding difficulties with the design of appropriate controllers. Different motion styles were generated by blending of the mixing weights w_{ij} and the trajectory mean values m_i.

2.3 Stepwise regression approach for the modeling of the individual actions

In order to model the step sequences with coordinated walking and reaching we approximated the training data by the described anechoic mixtures, using a step-wise regression approach that introduced different types of source functions for the three different component actions.

Reaching is a non-periodic movement and therefore requires the introduction of a non-periodic source function. In order to generate such a function online, the phase of the leading Hopf oscillator was derived from the state variables according to the relationship $\phi(t) = \mathrm{mod}_{2\pi}(\arctan(y(t)/x(t)))$, (ensuring $0 \leq \phi < 2\pi$). The non-periodic source signal was defined by $s_0(t) = \cos(\phi(t)/2)$, and the corresponding delay was set to zero.

The three actions of the training sequences were modeled as follows:

1st action: The weights of the non-periodic sources were determined in order to account for the non-periodic part of the training trajectory. Then this component was subtracted from the trajectory data, and the periodic source functions were determined by anechoic demixing, using an algorithm from [CdEG13], which had been modified in order to constrain all time delays belonging to the same source function to be equal. This constraint simplifies the blending between different motion styles, since then the delays of the sources are identical over styles, so that they do not have to be blended. Compared to the unconstrained anechoic model, this constraint requires the introduction of more sources for the same approximation quality (see Fig. 4). The first step could be modeled with sufficient accuracy using three periodic sources in addition to the non-periodic one.

2nd action: In order to model the second highly adaptive step, five periodic sources were required. The first three periodic sources were identical with the ones used for the approximation of the first action, and also the corresponding delays. The weights were optimized in order to minimize the remaining approximation error. The contributions of these three periodic sources (and of the non-periodic sources), then were subtracted from the training data, and two additional periodic sources were learned from the residuals (with constant delays across trials).

3rd action: In order to approximate this action, we used the same non-periodic and five periodic source signals, with the same time delays, that were identified for the modeling of the second action, while the weights of these sources were re-estimated.

The estimated source functions are shown in Fig. 5. The dotted curve illustrates the non-periodic source.

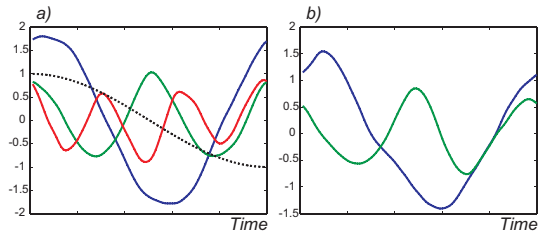


Figure 5: The source signals extracted by anechoic demixing algorithm. **a)**: three periodic source signals extracted from the first action and non-periodic source signal (dashed line). **b)**: two additional periodic source signals that were used for the modeling of the second and the third actions.

The source functions illustrated in the upper panel were used for the approximation of all three actions, and the two in the lower panel only for actions two and three.

Fig. 4 shows the approximation quality as a function of the number of source functions for the first and the second action, comparing normal anechoic demixing [OG11], our algorithm with constant delays over the different conditions [CdEG13], and a reconstruction using PCA. The measure for approximation quality was defined as $Q = 1 - (\|X - \hat{X}\|_F^2)/\|X\|_F^2$, where X is the matrix with the samples of the original signal, and $\hat{X}$ is the reconstructed signal, $\|\cdot\|_F^2$ is the squared Frobenius norm. Especially, the model without constraints for the delays still achieves significantly better approximation quality than PCA. The reconstruction error for the first action (purple circle on Fig. 4) is 95.6%, while the one with the two additional sources, used for actions 2 and 3, is 96.7% for the whole dataset (red circle).

The absolute values of the amplitudes of the weights for a single trajectory are depicted at Fig. 6, separately for the two source signals that carried the maximum amount of variance. This is the non-periodic source and the periodic source with the lowest frequency. The figure shows that the primitives clearly contribute to the different degrees of freedom of the human body. The non-periodic source primarily contributes to the joint angles of the arm, while the periodic source function strongly influences the hip and the leg joints. This clearly reflects the organization of human full body movements in terms of movements primitives. The figure also shows that the contribution of the sources changes between the steps. In the first action the contribution of the first periodic source is dominant, while in the second and last action the non-periodic source function makes a dominant contribution, reflecting the non-periodic reaching movement.

2.4 Online blending of the mixing weights

As illustrated in Fig. 6, the mixing weights change between the different actions within the sequence. For the modeling of a smooth transitions between the different actions the mixing weights thus had to be smoothly

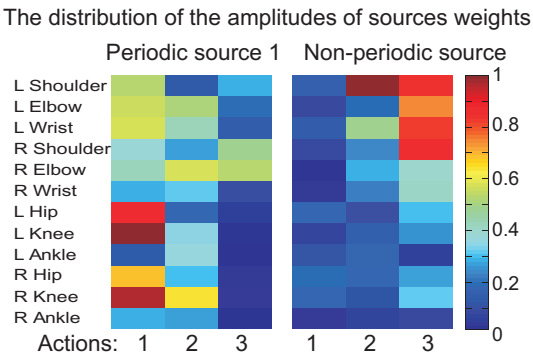


Figure 6: Absolute values of the weights for an example trajectory of the data set. The computed mixing weights are shown from the different actions within the sequence for the periodic source function with minimum frequency and for the non-periodic source. The color code is the same for both panels.

interpolated in an online fashion at the transitions between the individual actions.

For the weights associated with the periodic sources, the corresponding weight matrices were linearly blended according to the relationship $W(t) = (1 - \alpha(t))W_{\text{prev}} + \alpha(t)W_{\text{post}}$, where W_{prev} is the weight matrix in the step prior to the transition and W_{post} the one after the transition. The mean values for each of the angle trajectories were morphed accordingly: $m(t) = (1 - \alpha(t))m_{\text{prev}} + \alpha(t)m_{\text{post}}$, where m_{prev} is the mean value in the step prior to the transition and m_{post} is the one after the transition. The time-dependent blending weight $\alpha(t)$ was constructed from the phase variable $\phi(t)$ of the leading oscillator. Identifying the transition point, where the weights switch between the subsequent actions with the phase $\phi = 0$, the blending weight was given by the equation (here, regarding only two adjunct actions, we use convention: $\phi \in [-2\pi; 0[$ for a previous action, and $\phi \in [0; 2\pi[$ for a next one):

$$\alpha(t) = \left\{ \begin{array}{cc} 0 & \phi < -\beta, \\ (1 + \sin(\frac{\pi\phi(t)}{2\beta}))/2 & \phi \in [-\beta; \beta], \\ 1 & \phi > \beta \end{array} \right\} \quad (2)$$

The parameter $\beta = \pi/5$ determines the width of the interpolation interval and was chosen to guarantee natural-looking transitions. This value was derived in previous work, optimizing transitions for other scenarios [GMP+09].

The weights associated with the non-periodic source had to be treated separately since they can have different signs before and after the transition. Since the timing of this source is completely determined by the phase $\phi(t)$ of the leading oscillator, we constrained the blending by allowing sign changes for these weights only at the point where this phase crosses zero ($\phi(t) = 0$). The ramp-like non-periodic source is normalized in a way so that $s_0(0) = 1$ and $s_0(T) = -1$

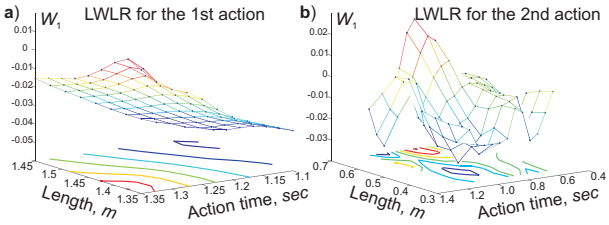


Figure 7: Learned nonlinear mappings between action length and duration and the mixing weight of the 1st source for hip flexion angle: **a)** 1st action, **b)** 2nd action.

(T being the duration of an oscillation of the leading oscillator in the attractor state). The following morphing rule $W(t) = \text{sign}(\phi(t))[(\alpha(t)-1)W_{prev} + \alpha(t)W_{post}]$ ensures a smooth transition that make the weights for this source converge at the boundaries between the actions against the value $\xi_{trans} = (m_{\text{prev}} + m_{\text{post}})/2 + (W_{\text{post}} - W_{\text{prev}})/2$.

2.5 Learning of mappings between step parameters and mixing weights

In order to make the generated behavior highly adaptive for conditions that were not in the training data and for dynamic changes of the environment, we devised an online control algorithm for the blending of the weights W, separately for each action. For this purpose, we learned nonlinear functions that map the step lengths and the duration of the steps onto the mixing weights. For the learning of this highly nonlinear mapping we used locally weighted linear regression (LWLR, [AMS97]). Fig. 7 shows some example for the weights of the first periodic source.

The required step lengths are computed online from the total distance to the drawer. The length of the step of the second action was optimized in order to generate an optimum (maximally comfortable) distance for the third action, which was estimated from the human data to be about $0.6m$. The total distance between the start position and the drawer D was then redistributed between the first two actions using a linear weighting scheme, specifying the relative contributions by the weight parameter γ. The remaining distance $D - 0.6\text{m}$ was then distributed according to the relationships $D_1 = (D - 0.6\text{m})\gamma$ and $D_2 = (D - 0.6\text{m})(1-\gamma)$, where we fitted $\gamma = 0.385$ based on the human data. This approach is motivated by the hypothesis that in humans predictive planning optimizes *end-state comfort*, i.e. the distance of the final reaching action [LRSS13].

We extended the algorithm in addition by a method that introduces additional normal steps (corresponding to action 1), in cases where the goal distance exceeds the distance that can be modeled without artifacts by a three-action sequence. If the distance between the goal and the agent was too short for the introduction of long steps, instead a variable number of short steps as in action 2 were introduced.

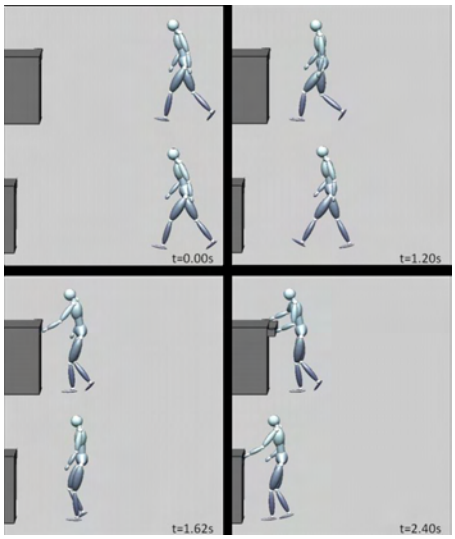


Figure 8: Two synthesized trajectories, illustrated in parallel for two conditions with different initial distance of the character from the drawer. Both animations look highly natural even though these goal distances were not present in the training data set.

3 RESULTS

Two example sequences of concatenated actions generated by our algorithm, for distances to the goal object that were not in the training set are shown in Fig. 8. An example video can be downloaded from [**Demo**[2]].

A more systematic evaluation shows that the algorithms can, without introducing additional steps, create natural looking coordinated sequences for goal distances between 2.34 and 2.94 m [**Demo**[3]]. If the specified goal distance exceeded this interval our system introduced automatically additional gait steps, making the system adaptive for goal distances beyond 3 meters. This is illustrated in [**Demo**[4]] that presents two examples of generated sequences for goal distances 3.84 and 4.62 m. With 3 actions the largest achievable range of goal distances without artifacts was about 60 cm, while adding another step increases this range to about 78 cm. Adding two or more normal gait steps our method is able to simulate natural-looking actions even for goal distances longer than 5 m. The next [**Demo**[5]] illustrates the sequence of three actions of first type followed by actions 2 and 3 for the goal distance 5.3 m.

Fig. 9 illustrates that, like in humans, the posture at the transition between the second and third action depends

[2] www.uni-tuebingen.de/uni/knv/arl/avi/wscg15/v2.avi
[3] www.uni-tuebingen.de/uni/knv/arl/avi/wscg15/v3.avi
[4] www.uni-tuebingen.de/uni/knv/arl/avi/wscg15/v4.avi
[5] www.uni-tuebingen.de/uni/knv/arl/avi/wscg15/v5.avi

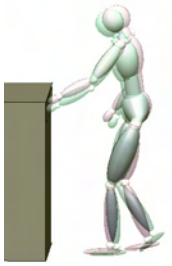


Figure 9: Postures at the transition between actions 2 and 3 for different lengths of the second action (red: 0.53 m , green: 0.39 m). Even though the distances to the drawer are the same in the last action the postures differ due to the predictive planning of the second action.

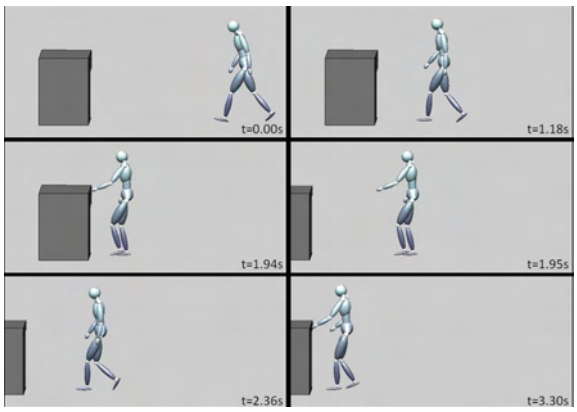


Figure 10: Online perturbation experiment. The goal (drawer) jumps away during the approaching of the character. The online planning algorithm introduces automatically an action of type 2 (short step) to adjust for the large distance to the goal.

on the previous step. In one case the step lengths for action 2 were $0.53m$ and $0.39m$, while the distance in the last step was identical ($0.6m$). This illustrates that in fact the posture for the reaching is modified in a predictive manner over multiple steps, where the predictive planning modifies the posture at the beginning of the last action even if the distance to the goal object for this action is identical. A planning scheme that is not predictive would predict here the same behaviors for the last action since the relevant control variable (distance from the object) is identical for both cases.

An even more extreme demonstration of this online adaptivity is shown in movie [**Demo**[6]]. Here the drawer jumps away during the approaching behavior by a large distance so that it can no longer be reached with the originally planned number of steps. (Fig. 10). The online planning algorithm adapts to this situation by automatically introducing an additional step so that the behavior is successfully accomplished. Again the behavior has a very natural appearance even though this scenario was not part of the training data set.

[6] www.uni-tuebingen.de/uni/knv/arl/avi/wscg15/v6.avi

4 CONCLUSIONS

We have presented a method for the online animation of multi-step human movements that was inspired by concepts derived from biological systems. The proposed system realizes a predictive planning of multi-step sequences, including periodic an non-periodic movements that reproduce critical properties observed in experiments on human motor planning. The planning is predictive and optimizes the 'comfort' during the execution of the final action. The proposed system exploits the concept of movement primitives in order to implement a flexible and highly natural-looking coordination of periodic and non-periodic behaviors of the upper and lower limbs, and to realize smooth transitions between subsequent actions within the sequence. For the first time, our architecture is implemented for generation of goal-directed movements. Our approach differs from the whole-body motion blending approach presented in [HK14], where, in order to increase naturalness of the transitions, it was necessary to introduce empirical rules that depend on the gait phase. Future work will extend our approach to other classes of movements, including, for instance, adaptive arm reaching movements accomplished while walking. In addition, we plan a systematic evaluation of the realism of the generated motions, including psychophysical studies.

ACKNOWLEDGEMENTS

The work supported by EC FP7 under grant agreements FP7-248311 (AMARSi), FP7-611909 (Koroibot), H2020 ICT-644727 (CogIMon), FP7-604102 (HBP), PITN-GA-011-290011 (ABC), DFG GI 305/4-1, DFG GZ: KA 1258/15-1, BMBF, FKZ: 01GQ1002A. Authors thank Biwei Huang for help with data segmentation and animation.

REFERENCES

[AFO03] O. Arikan, D.A. Forsyth, and J. F. O'Brien. Motion synthesis from annotations. *ACM Trans. on Graphics, SIGGRAPH '03*, 22(3):402–408, 2003.

[AMJ07] Y. Abe, Da Silva M., and Popović J. Multiobjective control with frictional contacts. *ACM SIGGRAPH/Eurograph. Symp. on Comp. Anim.*, 2007.

[AMS97] C. G. Atkeson, A. W. Moore, and S. Schaal. Locally weighted learning. *A.I. Review*, 11:11–73, 1997.

[Ber67] N.A. Bernstein. *The coordination and regulation of movements*. Pergamon Press, N.Y., Oxford, 1967.

[BRI06] J. Buchli, L. Righetti, and A. J. Ijspeert. Engineering entrainment and adaptation in limit cycle systems - from biological inspiration to applications in robotics. *Biol. Cyb.*, 95(6):645–664, 2006.

[CdEG13] E. Chiovetto, A. d'Avella, D. Endres, and M. A. Giese. A unifying algorithm for the identification of kinematic and electromyographic motor primitives. *Bernstein Conference*, 2013.

[CG13] E. Chiovetto and M. A. Giese. Kinematics of the coordination of pointing during locomotion. *Plos One*, 8(11), 2013.

[CL01] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[CMCH96] H. Carnahan, B. J. McFadyen, D. L. Cockell, and A. H. Halverson. The combined control of locomotion and prehension. *Neurosci. Res. Comm.*, 19:91–100, 1996.

[FH05] T. Flash and B. Hochner. Motor primitives in vertebrates and invertebrates. *Curr. Opin. Neurobiol.*, 15(6):660–666, 2005.

[FMJ02] A. Fod, M. J. Mataric, and O. C. Jenkins. Motor primitives in vertebrates and invertebrates. *Auton. Robots*, 12(1):39–54, 2002.

[FP03] A. Fang and N. S. Pollard. Efficient synthesis of physically valid human motion. *ACM Trans. on Graphics*, 22(3):417–426, 2003.

[FXS12] A. W. Feng, Y. Xu, and A. Shapiro. An example-based motion synthesis technique for locomotion and object manipulation. *Proc. of ACM SIGGRAPH I3D*, pages 95–102, 2012.

[GMP+09] M. A. Giese, A. Mukovskiy, A. Park, L. Omlor, and J. J. E. Slotine. Real-time synthesis of body movements based on learned primitives. In D. Cremers et al., editor, *Stat. and Geom. Appr. to Vis. Mot. Anal., LNCS5604*, pages 107–127. Springer, 2009.

[GRIL08] A. Gams, L. Righetti, A. J. Ijspeert, and J. Lenarcic. A dynamical system for online learning of periodic movements of unknown waveform and frequency. *Proc. of the IEEE RAS / EMBS Int. Conf. on Biomed. Robotics and Biomechatronics*, pages 85–90, 2008.

[GSKJ03] M. Gleicher, H. J. Shin, L. Kovar, and A. Jepsen. Snap-together motion: Assembling run-time animation. *ACM Trans. on Graphics, SIGGRAPH '03*, 22(3):702–702, 2003.

[HK14] Y. Huang and M. Kallmann. Planning motions for virtual demonstrators. In *Intelligent Virtual Agents*, pages 190–203. Springer, 2014.

[HPP05] E. Hsu, K. Pulli, and J. Popovic. Style translation for human motion. *ACM Trans. on Graphics*, 24:1082–1089, 2005.

[KGP02] L. Kovar, M. Gleicher, and F. Pighin. Motion graphs. *Proc. of SIGGRAPH 2002*, pages 473–482, 2002.

[LHP05] K. Liu, A. Hertzmann, and Z. Popović. Learning physics-based motion style with nonlinear inverse optimization. *ACM Trans. on Graphics*, 23(3):1071–1081, 2005.

[LRSS13] W. M. Land, D. A. Rosenbaum, S. Seegelke, and T. Schack. Whole-body posture planning in anticipation of a manual prehension task Prospective and retrospective effects. *Acta Psychologica*, 114:298–307, 2013.

[LS98] W. Lohmiller and J. J. E. Slotine. On contraction analysis for nonlinear systems. *Automatica*, 34(6):683–696, 1998.

[LWS02] Y. Li, T. Wang, and H.Y. Shum. Motion texture: A two level statistical model for character motion synthesis. *Proc. of SIGGRAPH 2002*, pages 465–472, 2002.

[Mai90] P.-G. Maillot. Using quaternions for coding 3D transformations. In A. S. Glassner, editor, *Graphic Gems*, pages 498–515. Academic Press, Boston, MA, 1990.

[MB01] R.G. Marteniuk and C. P. Bertram. Contributions of gait and trunk movement to prehension: Perspectives from world- and body centered coordinates. *Motor Control*, 5:151–164, 2001.

[MLPP09] U. Muico, Y. Lee, J. Popović, and Z. Popović. Contact-aware nonlinear control of dynamic characters. *ACM Trans. on Graphics*, 28(3):Art.No.81., 2009.

[OG11] L. Omlor and M. A. Giese. Anechoic blind source separation using wigner marginals. *J. of Machine Learning Res.*, 12:1111–1148, 2011.

[PMSG09] A. Park, A. Mukovskiy, J. J. E. Slotine, and M. A. Giese. Design of dynamical stability properties in character animation. *Proc. of VRIPHYS 09*, pages 85–94, 2009.

[PSS02] S.I. Park, H.J. Shin, and S.Y. Shin. On-line locomotion generation based on motion blending. *Proc. of the 2002 ACM SIGGRAPH/Eurographics Symp. on Comp. Animation*, pages 105–111, 2002.

[RCB98] C. Rose, M. Cohen, and B. Bodenheimer. Verbs and adverbs: Multidimensional motion interpolation. *IEEE Comp. Graphics and Appl.*, 18(5):32–40, 1998.

[RGBC96] C. Rose, B. Guenter, B. Bodenheimer, and M. Cohen. Efficient generation of motion transitions using space-time constraints. *Int. Conf. on Comp. Graph. and Interactive Techniques, Proc. ACM SIGGRAPH'96*, 30:147–154, 1996.

[Ros08] D. A. Rosenbaum. Reaching while walking: reaching distance costs more than walking distance. *Psych. Bull. Rev.*, 15:1100–1104, 2008.

[SHP04] A. Safonova, J. Hodgins, and N. Pollard. Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *ACM Trans. on Graphics*, 23(3):514–521, 2004.

[SMKB14] A. Shoulson, N. Marshak, M. Kapadia, and N.I. Badler. Adapt: The agent development and prototyping testbed. *IEEE Trans. on Visualiz. and Comp. Graphics (TVCG)*, 99:1–14, 2014.

[ST05] W. Shao and D. Terzopoulos. Artificial intelligence for animation: Autonomous pedestrians. *Proc. ACM SIGGRAPH '05*, 69(5-6):19–28, 2005.

[WFH08] J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(2):283–298, 2008.

[WP95] A. Witkin and Z. Popović. Motion warping. *Proc. ACM SIGGRAPH'95*, 29:105–108, 1995.

[WS10] M. Weigelt and T. Schack. The development of end-state comfort planning in preschool children. *Exper. Psych.*, 57(6):476–782, 2010.

[WZ10] C.-C. Wu and V. Zordan. Goal-directed stepping with momentum control. *ACM SIGGRAPH/Eurographics Symp. on Comp. Animation (SCA) 2010*, 2010.

Kinect-Based Gait Recognition Using Sequences of the Most Relevant Joint Relative Angles

Faisal Ahmed, Padma Polash Paul, Marina L. Gavrilova
Department of Computer Science
University of Calgary
Calgary, AB, Canada
{faahmed, pppaul, mgavrilo}@ucalgary.ca

ABSTRACT

This paper introduces a new 3D skeleton-based gait recognition method for motion captured by a low-cost consumer level camera, namely the Kinect. We propose a new representation of human gait signature based on the spatio-temporal changes in relative angles among different skeletal joints with respect to a reference point. A sequence of joint relative angles (JRA) between two skeletal joints, computed over a complete gait cycle, comprises an intuitive representation of the relative motion patterns of the involved joints. JRA sequences originated from different joint pairs are then evaluated to find the most relevant JRAs for gait description. We also introduce a new dynamic time warping (DTW)-based kernel that takes the collection of the most relevant JRA sequences from the train and test samples and computes a dissimilarity measure. The use of DTW in the proposed kernel makes it robust in respect to variable walking speed and thus eliminates the need of resampling to obtain equal-length feature vectors. The performance of the proposed method was evaluated using a Kinect skeletal gait database. Experimental results show that the proposed method can more effectively represent and recognize human gait, as compared against some other Kinect-based gait recognition methods.

Keywords

Gait recognition, Kinect v2, joint relative angle (JRA), DTW-kernel, motion analysis.

1 INTRODUCTION

Over the past ten years, biometric recognition and authentication has attracted a significant attention due to its potential applicability in social security, surveillance systems, forensics, law enforcement, and access control [1, 2]. A biometric system can be defined as a pattern-recognition system that can recognize individuals based on the characteristics of their physiology or behavior [3, 4]. Gait is one of the very few biometrics that can be recognized at a distance without any direct participation or cooperation of the user. Gait recognition involves identifying a person by analyzing his/her walking pattern. Since human locomotion is a complex and dynamic process that comprises movements of different body limbs and their interactions with the environment [5], disguising one's gait or imitating some other person's gait is quite difficult. As a result, gait recognition is particularly useful in crime scenes where other biometric traits (such as face or fingerprint) might be obscured intentionally [6]. The non-invasive nature and the ability to recognize individuals at a distance makes gait an attractive biometric modality in security and surveillance systems [7, 8]. In addition, gait analysis has many applications in virtual and augmented reality, 3D human body modeling and animation [9, 10], motion and video retrieval [11], health care [12]), etc.

In this paper, we present a new Kinect-based gait recognition method that exploits the relative motion patterns of different skeletal joints to represent the gait features. The proposed method encodes the relative motion between two joints by computing the joint relative angles (JRA) over a complete gait cycle. Here, JRA is defined as the angles formed by the corresponding two joints with respect to a reference point in a 3D space. Relevance of a particular joint pair in gait feature representation is then evaluated based on an intuitive statistical analysis that reflects the level of engagement of a particular joint pair in human walking. Finally, we introduce a new dynamic time warping (DTW)-based kernel, which is used to compute the dissimilarity between the collection of JRA sequences obtained from two gait samples. The performance of the proposed method is evaluated using a 20-person skeletal gait database captured using the Kinect v2 sensor. The experimental analysis shows that the proposed method can represent and recognize human gait in a more effective manner,

as compared against some existing Kinect-based gait recognition methods.

2 RELATED WORK

Different gait recognition methods found in literature can be divided into two categories: i) model-based approaches and ii) model-free approaches [13]. In model-based approaches, explicit models are used to represent human body parts (legs, arms, etc.) [14]. Parameters of these models are estimated in each frame and the change of the parametric values over time is used to represent gait signature. However, the computational cost involved with model construction, model fitting, and estimating parameter values makes most of the model-based approaches time-consuming and computationally expensive [14]. As a result, they are unsuitable for a wide range of real-world applications. One of the early parametric gait recognition methods was proposed by BenAbdelkader et al. [15], where they estimated two spatiotemporal parameters of gait, namely stride length and cadence as two distinctive biometric traits. Later, Urtasun and Fua [16] proposed a gait analysis method that relies on fitting 3-D temporal motion models to synchronized video sequences. Recovered motion parameters from the models are then used to characterize individual gait signature. A similar approach proposed by Yam et al. [17] models human leg structure and motion in order to discriminate between gait signatures obtained from walking and running. Although this method presents an effective way to view and scale independent gait representation, it is computationally expensive and sensitive to the quality of the gait sequences [18].

Instead of modeling individual body parts, the model-free approaches utilize the silhouette as a whole in order to construct a compact representation of walking motion [14]. Gait energy image (GEI) [19] and motion energy image (MEI) [20] are two of the most well-known model-free gait recognition methods. The basis of the MEI representation is a temporal vector image. Here, each vector point holds a value, which is a function of the motion properties at the corresponding sequence image [20]. On the other hand, GEI accumulates all the silhouette motion sequences in a single image, which preserves the temporal information as well [19]. Many of the recent model-free gait recognition methods extend GEI to a more robust representation. For example, Chen et al. [21] proposed frame difference energy image (FDEI), which utilizes denoising and clustering in order to suppress the influence of silhouette incompleteness. Li and Chen [22] fused foot energy image (FEI) and head energy image (HEI) in order to construct a more informative energy image representation. Although model-free approaches are computationally inexpensive, they are sensitive to view and scale changes and therefore, not suitable in uncontrolled environments.

While biometric gait recognition has been studied for the past twenty years, the recent popularization and low cost of Kinect has contributed to the spike in the interest in gait recognition using Kinect data. Kinect is a low-cost consumer-level device made up of an array of sensors, which includes i) a color camera, ii) a depth sensor, and iii) a multi-array microphone setup. Figure 1 shows different data streams that can be obtained from the Kinect. In addition, Kinect sensor can track and construct a 3D virtual skeleton from human body in real-time [23] (as shown in Figure 2), which renders the time consuming video processing steps unnecessary. All these functionalities of Kinect have led to its application in different real-world problems, such as home monitoring [24], health care [25], surveillance [26], etc. The low computation real-time skeleton tracking feature has encouraged some recent gait recognition methods that extract features from the tracked skeleton model. One of the pioneer studies conducted by Ball et al. [7] used Kinect for unsupervised clustering of gait samples. Features were extracted only from the lower body part. Preis et al. [27] presented a Kinect skeleton-based gait recognition method based on 13 biometric features: height, the length of legs, torso, both lower legs, both thighs, both upper arms, both forearms, step-length, and speed. However, these features are mostly static and represent individual body structure, while gait is considered to be a behavioral biometric, which is more related to the movement patterns of body parts during locomotion. Gabel et al. [28] used the difference in position of these skeleton points between consecutive frames as their feature. However, the proposed method was only evaluated for gait parameter extraction rather than person identification.

In this paper, we investigate Kinect-based gait recognition by the means of a new feature, namely the joint relative angle (JRA). The motivation is to capture the relative motion patterns of different joint pairs by examining how the corresponding relative angle between them varies over time. We also introduce an extension of the dynamic time warping (DTW) method, namely the DTW-based kernel that evaluates a collection of JRA sequences for the recognition task.

3 PROPOSED METHOD

The proposed new gait recognition method utilizes the 3D skeleton data obtained from the Kinect v2 sensor. Robustness to view and pose changes are the main advantages offered by the proposed method. Released in mid-July 2014, Kinect v2 offers a greater overall precision, responsiveness, and intuitive capabilities than the previous version [29]. The v2 sensor has a higher depth fidelity that enables it to see smaller objects more

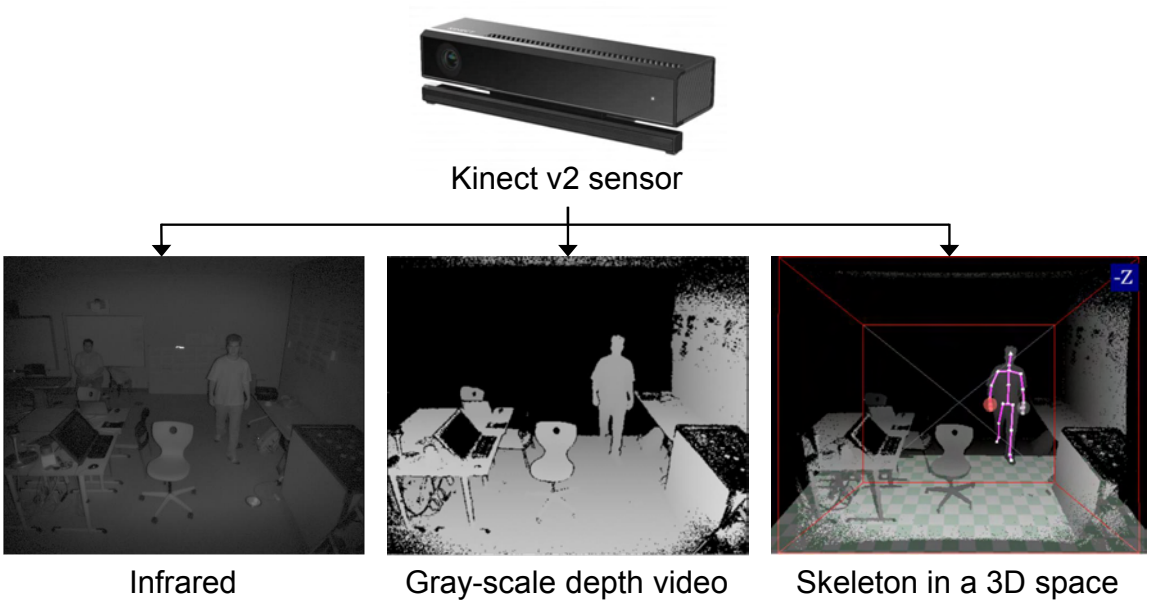
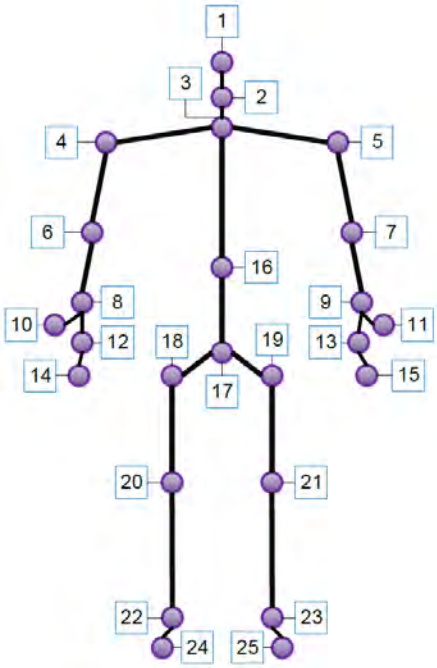


Figure 1: Different data streams obtained from the Kinect v2 sensor.

clearly, which results in a more accurate 3D object construction [29]. It can track a total of six people and 25 skeletal joints per person simultaneously [29]. In addition, while the skeleton tracking range is broader, the tracked joints are more accurate and stable than the previous version of the Kinect [29].

There are several steps involved in the proposed gait recognition method. The first step is to detect a complete gait cycle from the video sequence captured using the Kinect sensor. Since gait is a cyclic motion, detection of a complete gait cycle facilitates consistent feature extraction. Next, joint relative angle (JRA) features for different joint-pairs are computed over the complete gait cycle. One of the main advantages of using angle-based feature representation is that it is scale and view invariant. As a result, recognition is not constrained by a fixed distance from the camera or individuals walking only towards a specific direction in front of the camera. In order to assess the relevance of a particular JRA feature in gait representation, we employ a statistical analysis that evaluates the corresponding joint pair based on their involvement in gait movement. Only the most relevant joint pairs are considered in the proposed JRA-based gait feature representation. Once the feature representation is obtained, the proposed dynamic time warping (DTW)-based kernel is used for the classification task. The proposed kernel takes a collection of the most relevant JRA sequences from both the training and test samples as parameters and computes a dissimilarity measure between them. One particular advantage of the proposed kernel is that, it can match variable length JRA sequences originated due to variable walking speed in different videos of the same person,



1. Head	6. ElbowLeft	11. ThumbRight
2. Neck	7. ElbowRight	12. HandLeft
3. SpineShoulder	8. WristLeft	13. HandRight
4. ShoulderLeft	9. WristRight	14. HandTipLeft
5. ShoulderRight	10. ThumbLeft	15. HandTipRight

16. SpineMid	21. KneeRight
17. SpineBase	22. AnkleLeft
18. HipLeft	23. AnkleRight
19. HipRight	24. FootLeft
20. KneeLeft	25. FootRight

Figure 2: 3D skeleton joints tracked by the Kinect v2 sensor.

thus eliminating any need of pre-processing steps, such as resampling. Figure 3 shows the overview of the proposed gait recognition method.

3.1 Gait cycle detection

The first task of any gait recognition method is to isolate a complete gait cycle so that salient features can be extracted from it. Regular human walking is considered to be a cyclic motion, which repeats in a relatively stable frequency [14]. Therefore, features extracted from a single gait cycle can represent the complete gait signature. A gait cycle is composed of a complete cycle from rest (standing) position-to-right foot forward-to-rest-to-left foot forward-to rest or vice versa (left food forward followed by a right foot forward) [30]. In order to identify gait cycles, the horizontal distance between the AnkleLeft and AnkleRight joints was tracked over time, as shown in Figure 4. A moving average filter was used to smooth the distance vector. During the walking motion, the distance between the two ankle joints will be the maximum when the right and the left leg are farthest apart and will be the minimum when the legs are in the rest (standing) position. Therefore, by detecting three subsequent minima, it is possible to find the three subsequent occurrences of the two legs in the rest position, which corresponds to the beginning, middle, and ending points of a complete gait cycle, respectively [31].

3.2 Gait feature representation using joint relative angle (JRA)

The skeleton constructed by the Kinect v2 sensor comprises a hierarchy of 25 skeletal joints, where a connection between two joints forms a limb. Therefore, the raw data provided by the Kinect for gait is time series of 3D positions of these joints. However, this data lacks properties like invariance against view and scale changes, which makes direct use of this data as features infeasible. We present a new gait feature representation that processes this raw data and extracts the joint relative angles (JRA) formed by different pairs of joints with respect to a reference point. JRA between two joints p_1 and p_2 can be defined as the angle formed by p_1 and p_2 with respect to a reference point r. Given the coordinates of 3 points p_1, p_2, and r in a 3-D space, the angle Θ_{p_1,p_2} formed by $p_1 \rightarrow r \rightarrow p_2$ using the right hand rule from r can be calculated as:

$$\Theta_{p_1,p_2} = \cos^{-1} \frac{\overrightarrow{p_1r}.\overrightarrow{rp_2}}{||\overrightarrow{p_1r}||||\overrightarrow{rp_2}||} \quad (1)$$

Here, $\overrightarrow{p_1r} = r - p_1$, $\overrightarrow{rp_2} = p_2 - r$, the dot(.) represents dot product between two vectors, and $||\overrightarrow{p_1r}||$ and $||\overrightarrow{rp_2}||$ represent the length of $\overrightarrow{p_1r}$ and $\overrightarrow{rp_2}$, respectively. The SPINE_BASE joint was selected as the reference point, since it remains almost stationary during walking.

JRAs computed over time provide an intuitive representation of the relative movements of the joints involved. The advantages of using joint relative angle features are two-fold: firstly, the computed JRA features are view and scale independent. This means that, the feature values will not be affected by the variation of the distance of the subject from the camera or the direction of the subject's walking. Secondly, according to [7], joint distance-based features proposed in recent works [27], [28] are found to vary over time significantly. As a result, consistent feature extraction is difficult in some cases. On the other hand, although the distances of the joints vary over time, angles formed by the joints remain unaffected.

In this study, we consider JRAs originated from a particular joint-pair as a small fragment of a person's gait signature, where the full gait signature is defined as a collection of JRA sequences originated from different joint-pair combinations over a complete gait cycle. For the 25 skeletal joints, there is a total of 300 possible joint-pair combinations, which is a high-dimensional feature space. In addition, not all joint-pair is relevant in gait feature representation. For example, JRAs between the SpineShoulder and the SpineMid joints does not represent any information related to human gait, since both these joints remain almost stationary when a person walks. Therefore, identifying the skeletal joint-pairs that are relevant to human gait motion is imperative for the proposed gait recognition method.

3.3 Selection of the most relevant JRA sequences

Since not all skeletal joints engage during human locomotion, not all JRA features are relevant in gait representation. Relevance of a JRA sequence originated from a particular joint pair can be evaluated intuitively by analyzing human walking. In this paper, we present a statistics-based relevant joint pair selection approach, that utilizes histogram of JRA features to evaluate the level of engagement of the corresponding joint pair.

For joint pairs that has high relative motion during gait, the joint relative angles computed over the full gait cycle should have high temporal changes. On the other hand, joint pairs that remains stationary or moves little during gait should have little variation of JRA over the full gait cycle. This can also be represented using histogram of JRA values. For a particular joint pair that has high relative motion during gait, the histogram should have a wide distribution. On the other hand, for joint pairs that has little relative movement, the JRA values will occupy only a few number of bins in the histogram. Figure 5 shows histogram of JRA values computed for different joint pair combinations for 4 different participants. It can be observed that, for some joint pairs ({SpineShoulder, SpineMid},

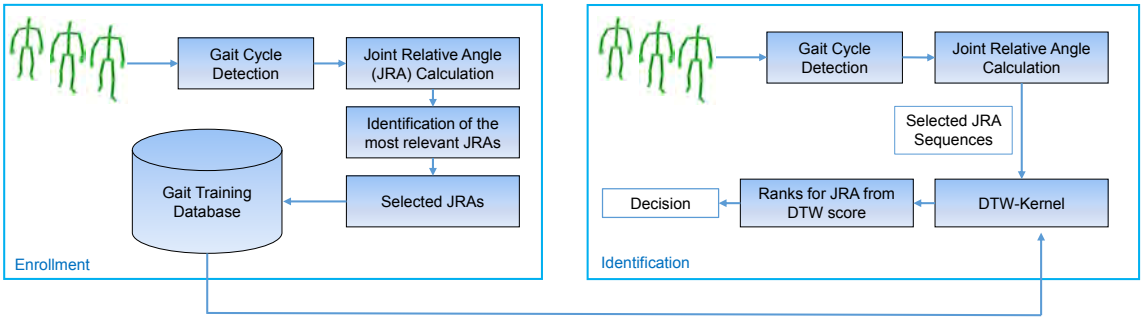


Figure 3: Overview of the proposed gait recognition method.

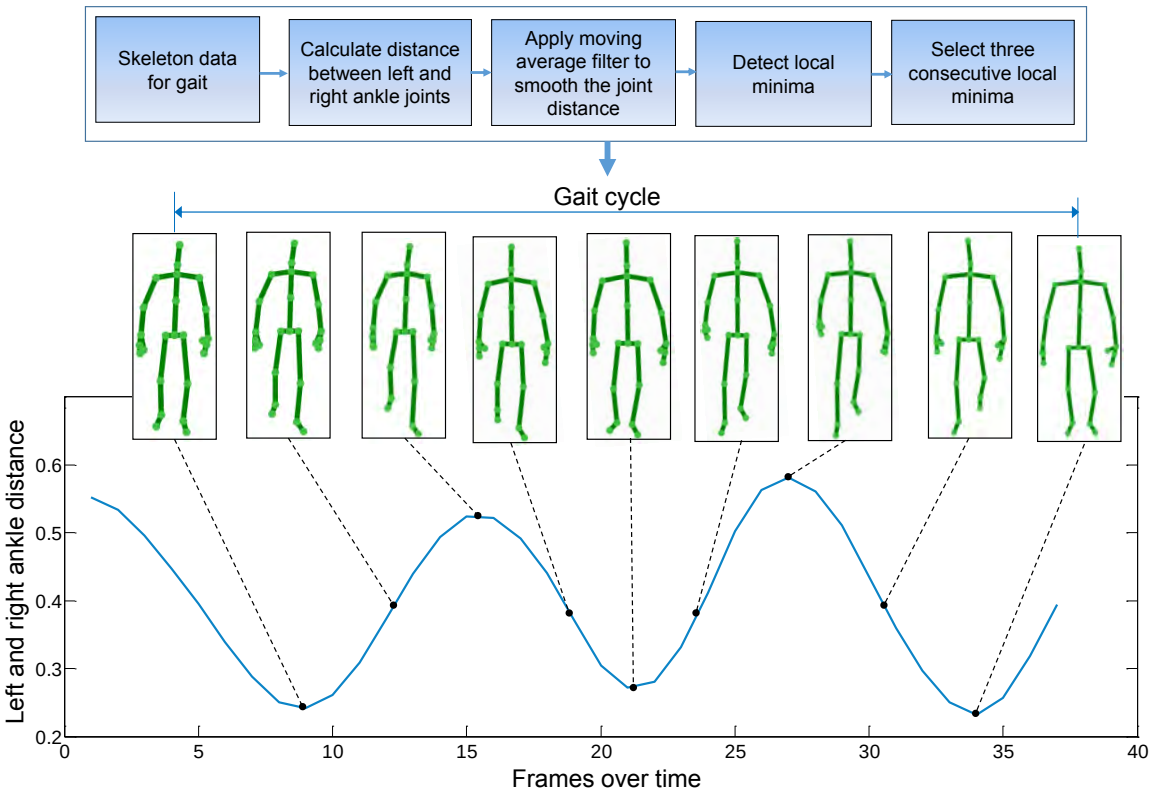


Figure 4: Detection of a complete gait cycle by tracking the distance between the left and right ankle joints.

{ShoulderLeft, ShoulderRight}, {HipLeft, HipRight}), the temporal change of JRA values over the complete gait cycle is really small and therefore, the distribution of JRA values in the histogram is really narrow (occupying only 2 or 3 bins). On the other hand, for joint pairs like {AnkleLeft, AnkleRight}, {ShoulderLeft, AnkleLeft}, and {ShoulderRight, AnkleRight}, the JRA values occupy a large number of bins in the histogram. Based on this observation, we argue that, the number of bins occupied in a JRA histogram of a particular joint pair is an important measure to quantify the level of engagement of the corresponding joint pair in human gait. This, in turn, quantizes the relevance of the corresponding joint pair in the gait movement. In this paper, we use the number of occupied bins in the JRA histogram of a particular joint pair to represent the relevance of that joint pair in gait feature representation. A high number of occupied bins represents a high relevance, while a small number represents a low relevance.

3.4 DTW-kernel for gait recognition

Joint relative angles (JRA) for different joint-pairs computed over a full gait cycle essentially represent sequences of time-series data. Alignment of such temporal gait data is a challenging task due to variation of walking speed, which might result in variable length

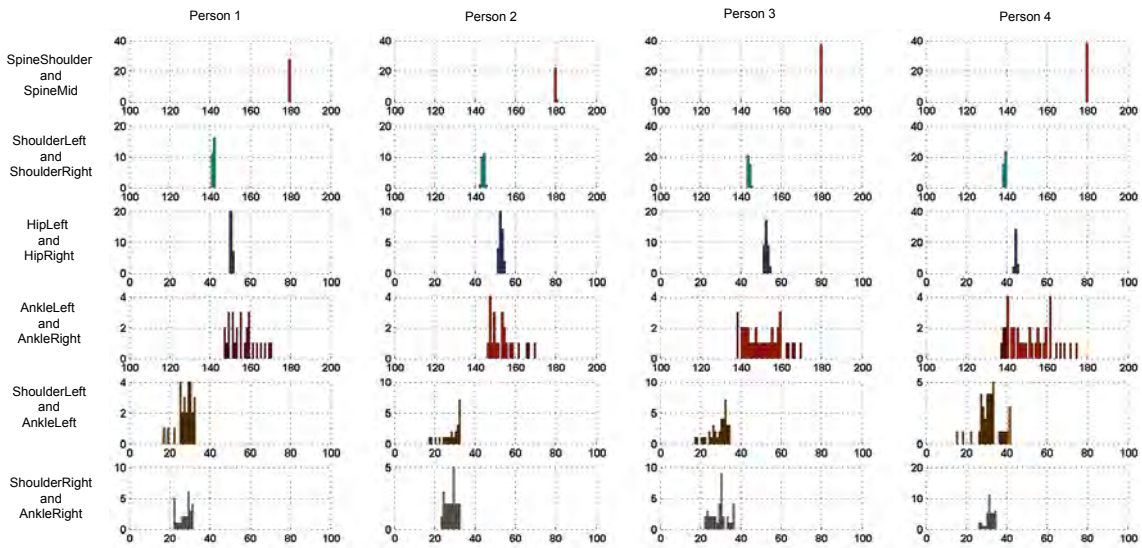


Figure 5: Histogram of JRA values for different joint pairs and persons. It can be observed that, some joint pairs have a wide distribution of JRA values in the histogram, while some other joint pair JRA values occupy only a small portion of the histogram bins.

JRA sequences for the same person. Therefore, applying traditional classifiers in this scenario requires extra pre-processing steps, such as resampling to obtain equal-length feature vectors. However, resampling of time-sequence data involves deletion or adding new data, which might affect the recognition performance. On the other hand, non-linear time sequence alignment techniques can effectively reduce the effect of variable walking speed by warping the time axis. Dynamic time warping (DTW) is a well-known non-linear sequence alignment technique. Originally proposed for speech signal alignment [32], recent DTW applications are mostly verification-oriented, such as offline signature verification [33]. In this paper, we propose to utilize DTW to design a kernel for gait recognition that takes a collection of JRA time series data originated from different joint pairs as the parameter and outputs the dissimilarity measure between two given gait samples. Use of DTW allows the alignment of different length JRA sequences, which enables to match gait samples without any intermediate resampling stage.

Given the set of all joint relative angles JRA = $\{\theta_1, \theta_2, ..., \theta_q\}$, where each θ_i represents JRAs for two particular joints with respect to the reference point computed over a full gait cycle, we first obtain a subset of the most relevant JRA sequences:

$$\theta = \{\theta_i | i = 1, 2, ..., M \quad where \quad \theta_i \in JRA\} \tag{2}$$

Let, θ_{train} and θ_{test} are two JRA sequences from the same joint-pair computed over a complete gait cycle, where the length of θ_{train} and θ_{test} are represented as $|\theta_{train}|$ and $|\theta_{test}|$, respectively.

$$\theta_{train} = a_1, a_2, a_3, ..., a_{|\theta_{train}|} \tag{3}$$

$$\theta_{test} = b_1, b_2, b_3, ..., b_{|\theta_{test}|} \tag{4}$$

Here, a_t and b_t are the JRA values of θ_{train} and θ_{test} at time t, respectively. Given these two time series, DTW constructs a warp path $W = w_1, w_2, w_3, ..., w_L$, where $max(|\theta_{train}|, |\theta_{test}|) \leq L \leq |\theta_{train}| + |\theta_{test}|$. Here, L is the length of the warp path between the two JRA sequences. Each element of the path can be represented as $w_l = (x, y)$, where x and y are two indices from the θ_{train} and θ_{test}, respectively. There are a number of constraints that DTW must satisfy. Firstly, the warp path must start at $w_1 = (1, 1)$ and end at $w_L = (|\theta_{train}|, |\theta_{test}|)$. This in turn ensures that, every index from the both time series is used in path construction. Secondly, if an index i from θ_{train} is matched with an index j from θ_{test}, it is prohibited to match any index $> i$ with any index $< j$ and vice-versa. This restricts the path from going back in time. Given these restrictions, the optimal warp path can be defined as the minimum distance warp path $dist_{optimal}(W)$:

$$dist_{optimal}(W) = min \sum_{l=1}^{L} \{dist(w_{li}, w_{lj})\} \tag{5}$$

Here, w_{li} and w_{lj} are two indices from θ_{train} and θ_{test}, respectively and $dist(w_{li}, w_{lj})$ is the Euclidean distance between w_{li} and w_{lj}.

We extend this basic DTW formulation to a kernel in order to compute the dissimilarity between a training and a testing gait sample, each of which is a collection of JRA sequences of different joint-pairs. The proposed DTW-kernel aligns the training and testing JRA

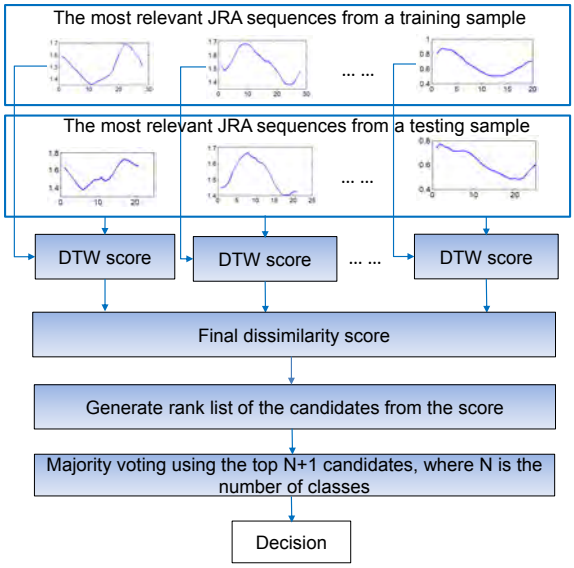


Figure 6: Proposed classification scheme based on the DTW-Kernel and the collection of the most relevant JRA sequences.

sequences of the same joint-pair with each other and computes a match score between them. Summation of all the match scores obtained from the different joint-pair JRA sequences from the training and testing samples is treated as the final dissimilarity measure. Formally, the proposed DTW kernel Δ for JRA-based gait representation can be defined as:

$$\Delta(\theta,\theta') = \sum_{m=1}^{M}\{min\sum_{l=1}^{L}\{dist(w_{m,li},w_{m,lj})\}\} \qquad (6)$$

Here, $\theta = \{\theta_1,\theta_2,...\theta_M\}$ and $\theta' = \{\theta'_1,\theta'_2,...,\theta'_M\}$ are collections of JRA sequences from M different joint-pairs and $min\sum_{l=1}^{L}\{dist(w_{m,li},w_{m,lj})\}$ represents the minimum warp path distance between the m-th joint pair JRAs of θ and θ'.

For the classification task, we first apply the DTW-kernel to compute the dissimilarity score and rank the candidates accordingly. We use this ranklist for a majority voting scheme where the top $N+1$ (N is the number of classes) candidates are considered. Figure 6 illustrates the proposed method.

4 EXPERIMENTS AND RESULTS

4.1 Experimental setup and dataset description

The performance of the proposed method is evaluated using a Kinect skeletal gait database, provided by the SMART Technologies, Calgary, Canada. The gait database comprises 20 participants (14 male, 6 female), from around 20 to 35 years old. For each person, a series of 3 videos was recorded in a meeting room environment. The position of the Kinect was fixed throughout the recording session. Each of the video scenes contains a participant entering the meeting room, walking toward a chair, and then sitting on the chair. Figure 7 shows a frame of a sample video from the gait database. We conducted a 3-fold cross-validation in order to evaluate the effectiveness of the proposed method. In a 3-fold cross-validation, the whole dataset is randomly divided into 3 subsets, where each subset contains an equal number of samples from each category. The classifier is trained on 2 subsets, while the remaining one is used for testing. The average classification rate is calculated after repeating the above process for 3 times. Since the database comprises 3 videos per person, in each fold, two videos were used for the training and the remaining one was used for testing.

4.2 Results and Discussions

The first step in our experimental analysis is to detect the most relevant joint pairs in order to represent the gait. For this purpose, we use the methodology proposed in section 3.3. For the 25 skeletal joints tracked by the Kinect v2 sensor, we construct a 25×25 matrix for each video sequence, where each cell corresponds to the number of bins occupied in the histogram of JRA values for a particular joint pair. Since our database comprises 20 participants and 3 videos per participant, we obtain a total of 60 matrices. For further analysis, we compute the average matrix from the 60 matrices. A heat map of the obtained 25×25 average matrix is shown in Figure 8. The heat map is symmetric on the both side of the diagonal, since the JRA values beween joints {J1, J2} and {J2,J1} are same. This map provides a comprehensive representation of the relevance of a particular joint pair in gait representation, where high value corresponds to high relevance and low value corresponds to a low relevance.

Based on this representation of joint pair relevance, we select subsets of JRA sequences for different thresholds and evaluate the recognition performance. For a threshold value of t, only the joint pair combinations with at least t bins occupied in the JRA histogram were selected for feature representation. Figure 9 shows the recognition performance of the proposed method for different subsets of JRA sequences selected for different threshold values. It can be observed that, increasing the number of bins excludes some of the less relevant joint pairs in the classification task, thus increasing the recognition performance. The highest recognition rate of 93.3% is obtained for JRA sequences that occupy more than or equal to 20 bins in the corresponding JRA histogram. Increasing the number of selected bins further results in a sharp decrease in the recognition perfor-

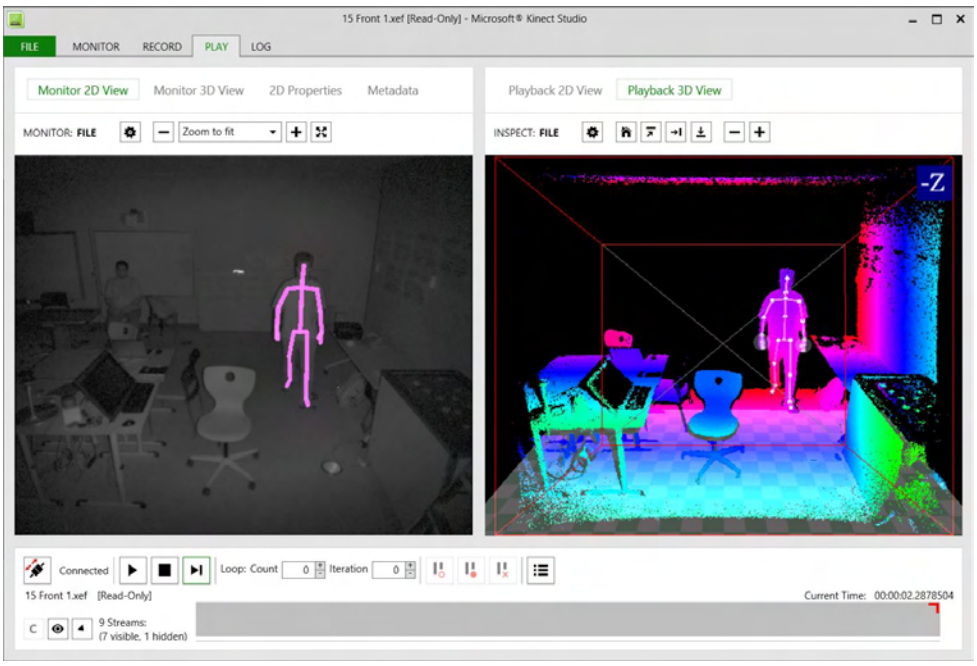


Figure 7: Sample video frame from the gait database captured using Kinect v2 sensor.

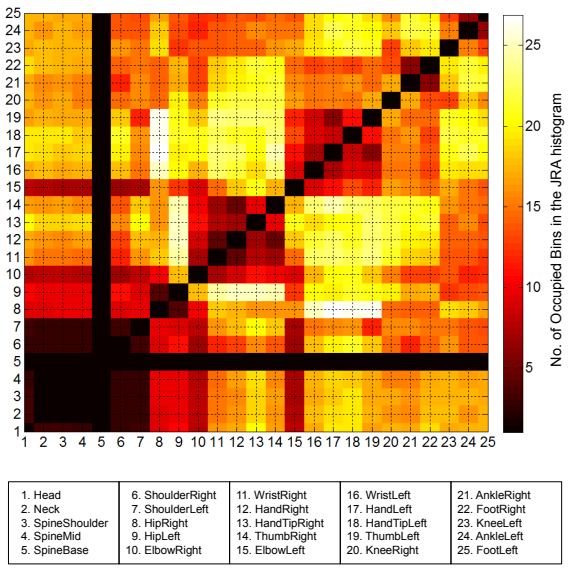


Figure 8: Heat map of the 25×25 average matrix obtained for the average number of bins occupied for different JRA histograms for all participants. Here, each point (i, j) represents the average number of occupied bins in the JRA histogram obtained for joint pair $\{i, j\}$.

mance. For the number of occupied bins > 20, Figure 10 shows a heat map representation of the selected joint pairs. Here, the dark points correspond to the excluded joints, while points with high heat corresponds to a relevant joint pair. This map is also symmetric. Therefore,

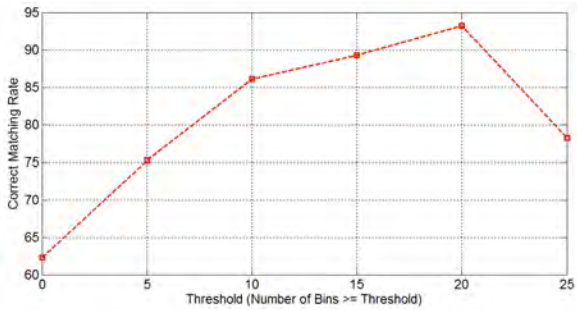


Figure 9: Performance of the most relevant JRA-based gait recognition for different number of occupied bins. The correct matching rate is obtained from 3-fold cross-validation.

only considering upper left triangle or lower right triangle formed by the diagonal (line from (1, 1) to (25, 25)) should be considered.

Finally, we compare the performance of the proposed method against some recent Kinect skeleton-based gait recognition methods. We have selected two studies and tested their performance on our gait database. Details of the selected two methods can be found in [7] and [27]. Table 1 shows the recognition performance of these methods. From the experimental results, it can be said that, gait recognition based on the collection of JRA sequences and DTW-kernel is more robust and achieves higher recognition performance than some of the existing gait recognition methods. The superiority of the proposed method is due to the utilization of view

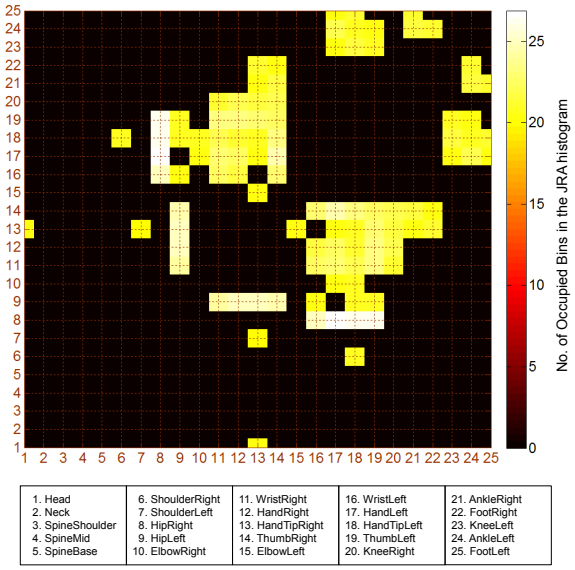


Figure 10: Heat map for the most relevant joint pair combinations found in our experiments. The dark region corresponds to the all joint pair combinations that are excluded from the final feature representation.

and pose invariant relative angle features coupled with a relevance evaluation and non-linear alignment of variable length feature sequences using the DTW-kernel.

Method	**Recognition Rate (%)**
Collection of the most relevant JRA sequence + DTW-Kernel	93.3
Ball et al. [7]	66.7
Preis et al. [27]	84.2

Table 1: Recognition rates of different methods for 3-fold cross-validation.

5 CONCLUSION

This paper presented a new Kinect-based gait recognition method that utilizes the 3D skeleton data in order to compute a robust representation of gait. We introduced a new feature, namely the joint relative angle that encodes the relative motion patterns of different skeletal joint pairs by computing the relative angles between them with respect to a reference point. To evaluate the relevance of a particular JRA sequence in gait feature representation, we constructed histograms of JRA features that can effectively be used to quantize the level of engagement of different joint pairs in human walking. Finally, we propose a dynamic time warping (DTW)-based kernel that takes the collection of the most relevant JRA sequences from both the train and test samples as parameters and computes a dissimilarity measure. Here, the use of DTW makes the proposed kernel robust against variable walking speed and thus eliminates any need of extra pre-processing. Experiments using a Kinect skeletal gait database showed excellent recognition performance for the proposed method, compared against some recent Kinect-based gait recognition methods. In the future, we plan to extend the proposed method for action recognition and motion retrieval.

6 ACKNOWLEDGMENTS

The authors would like to thank NSERC DISCOVERY program, URGC, NSERC ENGAGE, AITF, and SMART Technologies, Canada for partial support of this project.

7 REFERENCES

[1] Wang, C., Zhang, J., Wang, L., Pu, J., and Yuan, X. Human Identification Using Temporal Information Preserving Gait Template. IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 34, No. 11, pp. 2164-2176, 2012.

[2] Paul, P.P., Gavrilova, M. A Novel Cross Folding Algorithm for Multimodal Cancelable Biometrics. Intl. Jour. of Software Science and Computational Intelligence, Vol. 4, No. 3, pp. 20-37, 2012.

[3] Prabhakar, S., Pankanti, S., Jain, A.K. Biometric recognition: security and privacy concerns. IEEE Security and Privacy, Vol. 1, No. 2, pp. 33-42, 2003.

[4] Paul, P.P., Gavrilova, M. Multimodal Cancelable Biometrics. IEEE Intl. Conf. on Cognitive Informatics & Cognitive Computing, pp. 43-49, 2012

[5] Tanawongsuwan, R., Bobick, A. Gait Recognition from Time-normalized Joint-angle Trajectories in the Walking Plane. IEEE Conf. on Computer Vision and Pattern Recognition, Vol. 2, pp. 726-731, 2001.

[6] Nixon, M.S., Carter, J.N., Grant, M.G., Gordon L., Hayfron-Acquah, J.B. Automatic recognition by gait: progress and prospects Sensor Review, Vol. 23, No. 4, pp. 323-331, 2003.

[7] Ball, A., Rye, D., Ramos, F., Velonaki, M. Unsupervised Clustering of People from Ŝkeleton' Data. ACM/IEEE Intl. Conf. on Human Robot Interaction, pp. 225-226, 2012.

[8] Ahmed, F., Paul, P.P., Gavrilova, M.L. DTW-based kernel and rank level fusion for 3D gait recognition using Kinect. The Visual Computer, Springer, 2015 [Published Online, DOI:10.1007/s00371-015-1092-0].

[9] Zhang, Y., Zheng, J., Magnenat-Thalmann, N. Example-guided anthropometric human body

modeling. The Visual Computer (CGI 2014), pp. 1-17, 2014.

[10] Bae, M.S., Park, I.K. Content-based 3D model retrieval using a single depth image from a low-cost 3D camera. The Visual Computer (CGI 2013), Vol. 29, pp. 555-564, 2013.

[11] Zhou, L., Zhiwu, L., Leung, H., Shang, L. Spatial temporal pyramid matching using temporal sparse representation for human motion retrieval. The Visual Computer (CGI 2014), Vol. 30, pp. 845-854, 2014.

[12] Barth, J, Klucken, J, Kugler, P, Kammerer, T, Steidl, R, Winkler, J, Hornegger, J, Eskofier, B. Biometric and mobile gait analysis for early diagnosis and therapy monitoring in Parkinson's disease. Intl. Conf. of the IEEE Engg. in Medicine and Biology Society, pp. 868-871, 2011.

[13] Han, J., Bhanu, B. Statistical Feature Fusion for Gait-based Human Recognition. IEEE Conf. on Computer Vision and Pattern Recognition, Vol. 2, pp. 842-847, 2004.

[14] Wang, J., She, M., Nahavandi, S., Kouzani, A. A review of vision-based gait recognition methods for human identification. IEEE Intl. Conf. on Digital Image Computing : Techniques and Application, pp. 320-327, 2010.

[15] BenAbdelkader, C., Cutler, R., Davis, L. Stride and cadence as a biometric in automatic person identification and verification. IEEE Intl. Conf. on Automatic Face and Gesture Recognition, pp. 372-377, 2002.

[16] Urtasun, R., Fua, P. 3D Tracking for Gait Characterization and Recognition. IEEE Intl. Conf. on Automatic Face and Gesture Recognition, pp. 17-22, 2004.

[17] Yam, C., Nixon, M.S., Carter, J.N. Automated person recognition by walking and running via model-based approaches. Pattern Recognition, Vol. 37, pp. 1057-1072, 2004.

[18] Sinha, A., Chakravarty, K., Bhowmick, B. Person Identification using Skeleton Information from Kinect. Intl. Conf. on Advances in Computer-Human Interactions, pp. 101-108, 2013.

[19] Han, J., Bhanu, B. Individual recognition using gait energy image. IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 28, pp. 316-322, 2006.

[20] Bobick, A.F., Davis, J.W. The recognition of human movement using temporal templates. IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 23, pp. 257-267, 2001

[21] Chen, C., Liang, J., Zhao, H. Frame difference energy image for gait recognition with incomplete silhouettes. Pattern Recognition Letters, Vol. 30, pp. 977-984, 2009

[22] Li, X., Chen, Y. Gait Recognition Based on Structural Gait Energy Image. Jour. of Computational Information Systems, Vol. 9, No. 1, pp. 121-126, 2013

[23] Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-time human pose recognition in parts from single depth image. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1297-1304, 2011.

[24] Stone, E.E., Skubic, M. Evaluation of an inexpensive depth camera for passive in-home fall risk assessment. Intl. Pervasive Computing Technologies for Healthcare Conf., pp. 71-77, 2011

[25] Chang, Y.J., Chen, S.F., Huang, J.D.: A Kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities. Research in Developmental Disabilities, Vol. 32, No. 6, pp. 2566-2570, 2011.

[26] Popa, M., Koc, A.K., Rothkrantz, L.J.M., Shan, C., Wiggers, P.: Kinect Sensing of Shopping Related Actions. Communications in Computer and Information Science, Vol. 277, pp. 91-100, 2012.

[27] Preis, J., Kessel, M., Linnhoff-Popien, C., Werner, M. Gait Recognition with Kinect. Workshop on Kinect in Pervasive Computing, 2012.

[28] Gabel, M., Gilad-Bachrach, R., Renshaw, E., Schuster, A. Full body gait analysis with Kinect. Annual Intl. Conf. of the IEEE Engg. in Medicine and Biology Society, pp. 1964-1967, 2012.

[29] Kinect for windows features. Online, Available: http://www.microsoft.com/en-us/kinectforwindows/meetkinect/features.aspx, Accessed on February 20, 2015.

[30] Kale, A., Sundaresan, A., Rajagopalan, A.N., Cuntoor, N.P., Roy-Chowdhury, A.K., Kruger, V., Chellapa, R. Identification of Humans Using Gait. IEEE Trans. on Image Processing, Vol. 13, No. 9, pp. 1163-1173, 2004.

[31] Sarkar, S. et al. The humanID gait challenge problem: data sets, performance, and analysis. IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 27, No. 2, pp. 162-177, 2005.

[32] Kruskal, J.B., Liberman, M. The symmetric time-warping problem: from continuous to discrete. Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparisons. Addison-Wesley, Massachusetts, 1983.

[33] Shanker, A.P., Rajagopalan, A.N. Off-line signature verification using DTW. Pattern Recognition Letters, Vol. 28, pp. 1407-1414, 2007.

BoneSplit - A 3D Texture Painting Tool for Interactive Bone Separation in CT Images

Johan Nysjö, Filip Malmberg, Ida-Maria Sintorn, and Ingela Nyström
Centre for Image Analysis, Dept. of Information Technology,
Uppsala University, Sweden
{johan.nysjo,filip.malmberg,ida.sintorn,ingela.nystrom}@it.uu.se

ABSTRACT

We present an efficient interactive tool for separating collectively segmented bones and bone fragments in 3D computed tomography (CT) images. The tool, which is primarily intended for virtual cranio-maxillofacial (CMF) surgery planning, combines direct volume rendering with an interactive 3D texture painting interface to enable quick identification and marking of individual bone structures. The user can paint markers (seeds) directly on the rendered bone surfaces as well as on individual CT slices. Separation of the marked bones is then achieved through the random walks segmentation algorithm, which is applied on a graph constructed from the collective bone segmentation. The segmentation runs on the GPU and can achieve close to real-time update rates for volumes as large as 512^3. Segmentation editing can be performed both in the random walks segmentation stage and in a separate post-processing stage using a local 3D editing tool. In a preliminary evaluation of the tool, we demonstrate that segmentation results comparable with manual segmentations can be obtained within a few minutes.

Keywords

Bone Segmentation, CT, Volume Rendering, 3D Painting, Random Walks, Segmentation Editing

1 INTRODUCTION

Cranio-maxillofacial (CMF) surgery to restore the facial skeleton after serious trauma or disease can be both complex and time-consuming. There is, however, evidence that careful virtual surgery planning can improve the outcome and facilitate the restoration [27]. In addition, virtual surgery planning can lead to reduced time in the operating room and thereby reduced costs.

Recently, a system for planning the restoration of skeletal anatomy in facial trauma patients (Figure 1) has been developed within our research group [25]. As input, the system requires segmented 3D computed tomography (CT) data from the fractured regions, in which individual bone fragments are labeled. Although a collective bone segmentation can be obtained relatively straightforward by, for instance, thresholding the CT image at a Hounsfield unit (HU) value corresponding to bone tissue, separation of individual bone structures is typically a more difficult and time-consuming task. Due to bone tissue density variations and image imprecisions such as noise and partial volume effects, adjacent bones and bone fragments in a CT image are typically connected to each other after thresholding, and cannot be separated by simple connected component analysis or morphological operations. In the current procedure, the bones are separated manually, slice by slice, using the brush tool in the ITK-SNAP software [30]. This process takes several hours to complete and is the major bottleneck in the virtual surgery planning procedure.

Figure 1: Example of a patient who has suffered complex fractures on the lower jaw and the cheekbone. The individual bone fragments in the CT image have been segmented with our interactive 3D texture painting tool to enable virtual planning of reconstructive surgery.

1.1 Contribution

Here, we present an efficient interactive tool for separating collectively segmented bones and bone fragments in CT volumes. Direct volume rendering combined with an interactive 3D texture painting interface enable the user to quickly identify and mark individual bone structures in the collective segmentation. The user can paint markers (seeds) directly on the rendered bone surfaces as well as on individual CT slices. Separation of marked bones is then achieved through the random walks segmentation algorithm [12]. A local 3D editing tool can be used to refine the result. In a preliminary evaluation of the bone separation tool, we demonstrate that segmentation results comparable with manual segmentations can be obtained within a few minutes.

1.2 Related Work

Model-based segmentation techniques have been used for automatic segmentation of individual intact bones such as the femur and tibia, but are not suitable for segmentation of arbitrarily shaped bone fragments. Automatic bone segmentation methods without shape priors have been proposed [10][18][2] but are not general enough for fracture segmentation.

Manual segmentation can produce accurate results and is often used in surgery planning studies. However, it is generally too tedious and time-consuming for routine clinical usage, and suffers from low repeatability. Another problem with manual segmentation is that the user only operates at a single slice at the time and thus may not perceive the full 3D structure. This tends to produce irregular object boundaries.

Semi-automatic or interactive segmentation methods combine imprecise user input with exact algorithms to achieve accurate and repeatable segmentation results. This type of methods can be a viable option if automatic segmentation fails and a limited amount of user-interaction time can be tolerated to ensure accurate results. An example of a general-purpose interactive segmentation tool is [6]. Liu et al. [22] used a graph cut-based [4] technique to separate collectively segmented bones in the foot, achieving an average segmentation time of 18 minutes compared with 1.5–3 hours for manual segmentation. Fornaro et al. [9] and Fürnstahl et al. [11] combined graph cuts with a bone sheetness measure [7] to segment fractured pelvic and humerus bones, respectively. Mendoza et al. [23] adapted the method in [22] for segmentation of cranial regions in craniosynostosis patients. The TurtleMap 3D livewire algorithm [16] produces a volumetric segmentation from a sparse set of user-defined 2D livewire contours, and have been applied for segmentation of individual bones in the wrist. It is, however, not suitable for segmentation of thin bone structures such as those in the facial skeleton. Segmentation of individual wrist bones has also been investigated in [15][24].

In all the semi-automatic methods listed above, the user interacts with the segmentation via 2D slices. A problem with using slice-based interaction for bone segmentation is that it can be difficult to identify, mark, and inspect individual bone structures and contact surfaces, particularly in complex fracture cases.

Texture painting tools [17][29] enable efficient and intuitive painting of graphical models (3D meshes) via standard 2D mouse interaction. Mouse strokes in screen space are mapped to brush strokes in 3D object space. Mesh segmentation methods [20] utilize similar sketch-based interfaces for semi-automatic labeling of individual parts in 3D meshes. Bürger et al. [5] developed a direct volume editing tool that can be used for manual labeling of bone surfaces in CT images. Our proposed 3D texture painting interface extends this concept to semi-automatic segmentation.

2 METHODS

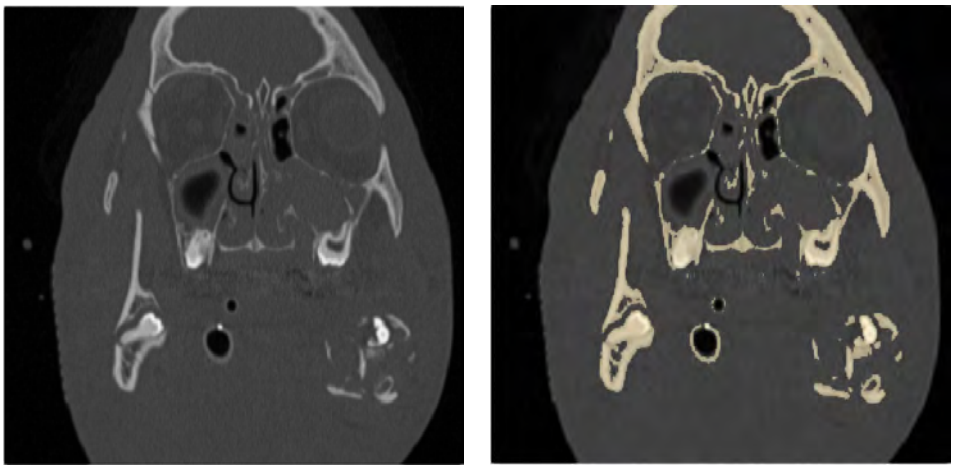
Our bone separation tool combines and modifies several image analysis and visualization methods, which are described in the following sections. In brief, the main steps are (1) collective bone segmentation, (2) marking of individual bone structures, (3) random walks bone separation, and (4) segmentation editing.

2.1 Collective Bone Segmentation

A collective bone segmentation is obtained by thresholding the grayscale image at the intensity value t_{bone} (see Figure 2). The threshold is preset to 300 HU in the system, but can be adjusted interactively, if needed, to compensate for variations in bone density or image quality. The preset value was determined empirically and corresponds to the lower HU limit for trabecular (spongy) bone. Noisy images can be smoothed with a $3 \times 3 \times 3$ Gaussian filter ($\sigma = 0.6$) prior to thresholding. The Gaussian filter takes voxel anisotropy into account and can be applied multiple times to increase the amount of smoothing, although usually a single pass is sufficient. Both the thresholding filter and the Gaussian filter utilize multi-threading to enable rapid feedback.

2.2 Deferred Isosurface Shading

We use GPU-accelerated ray-casting [19] to render the bones as shaded isosurfaces. The isovalue is set to t_{bone}, so that the visual representation of the bones matches the thresholding segmentation. Similar to [14] and [13], we use a deferred isosurface shading pipeline. A 32^3 min-max block volume is used for empty-space skipping and rendering of the ray-start positions (Figure 3a). We render the first-hit positions (Figure 3b) and surface normals (Figure 3c) to a G-buffer via multiple render



(a) (b)

Figure 2: Left: Coronal slice of a grayscale CT volume of the facial skeleton. Right: Collective bone segmentation obtained by thresholding the CT volume at a Hounsfield unit (HU) value corresponding to trabecular bone.

targets (MRT), and calculate shadows and local illumination in additional passes. Segmentation labels are stored in a separate 3D texture and fetched with nearest-neighbor sampling in the local illumination pass.

Local illumination (Figure 3e) is calculated using a normalized version of the Blinn-Phong shading model [1]. To make it easier for the user to perceive depth and spatial relationships between bones and bone fragments, we combine the local illumination with shadow mapping to render cast shadows (Figure 3f). The shadow map (Figure 3d) is derived from an additional first-hit texture rendered from a single directional light source's point of view. The shadows are filtered with percentage closer filtering (PCF) [26] and Poisson disc sampling to simulate soft shadows. It is possible to disable the shadows temporarily during the segmentation if they obscure details of interest.

Ambient lighting is provided from pre-filtered irradiance and radiance cube maps [1]. Unlike the traditional single-color ambient lighting commonly used in medical visualization tools, the color and intensity variations in the image-based ambient lighting allow the user to see the shape and curvature of bone structures that are in shadow. The image-based ambient lighting also enables realistic rendering of metallic surfaces, e.g., metallic implants that have been separated out from the bones as part of the planning procedure. To enhance fracture locations, we modulate the ambient lighting with a local ambient occlusion [21] factor, which is computed on-the-fly using Monte-Carlo integration.

2.3 3D Texture Painting Interface

As stated in Section 1.2, a problem with 2D slice-based interaction is that it may be difficult to identify, mark, and inspect individual bone structures. Even radiologists, who are highly skilled at deriving anatomical 3D structures from stacks of 2D images, may find it difficult to locate and mark individual bone fragments in complex fracture cases. To overcome this issue, we implemented a 3D texture painting interface that enables the user to draw seeds directly on the bone surfaces.

Our 3D brush (Figure 4a) is implemented as a spherical billboard and uses the first-hit texture (Figure 3b) for picking and seed projection. The brush proxy follows the bone surface and can only apply seeds on surface regions that are visible and within the brush radius (in camera space). To prevent the brush from leaking through small gaps in the surface of interest, we compute a local ambient occlusion term from a depth map derived from the first-hit texture, and discard brush strokes in areas where the ambient occlusion value at the brush center exceeds a certain threshold. The radius of the ambient occlusion sampling kernel corresponds to the radius of the brush.

Additional tools include a label picker, an eraser, a floodfill tool, and a local editing tool (Section 2.6). A 3D slice viewer enables the user to mark occluded bones or place additional seeds inside the bones. The latter can be useful when the boundaries between the bones are weak or when the image is corrupted by streak artifacts from metal implants. We also provide interactive clipping tools that can be used to expose bones and contact surfaces. Both the 3D slice viewer and the clipping tools are useful during visual inspection and editing of the segmentation result.

2.4 Random Walks Bone Separation

Given the collective binary bone segmentation, the next step is to separate the individual bones and bone fragments. We considered two graph-based segmentation algorithms, graph cuts [4] and random walks [12], for this task. In the end, we selected the random walks

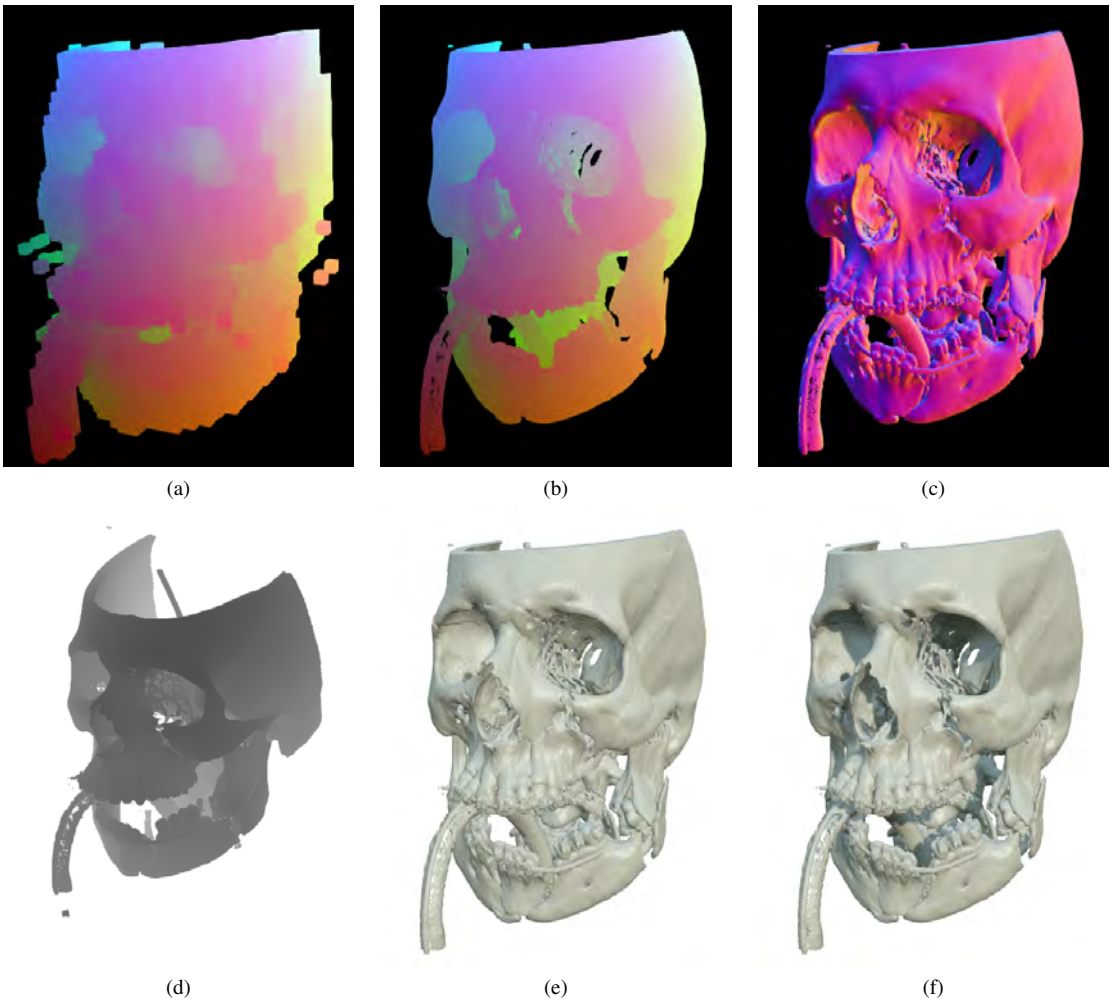


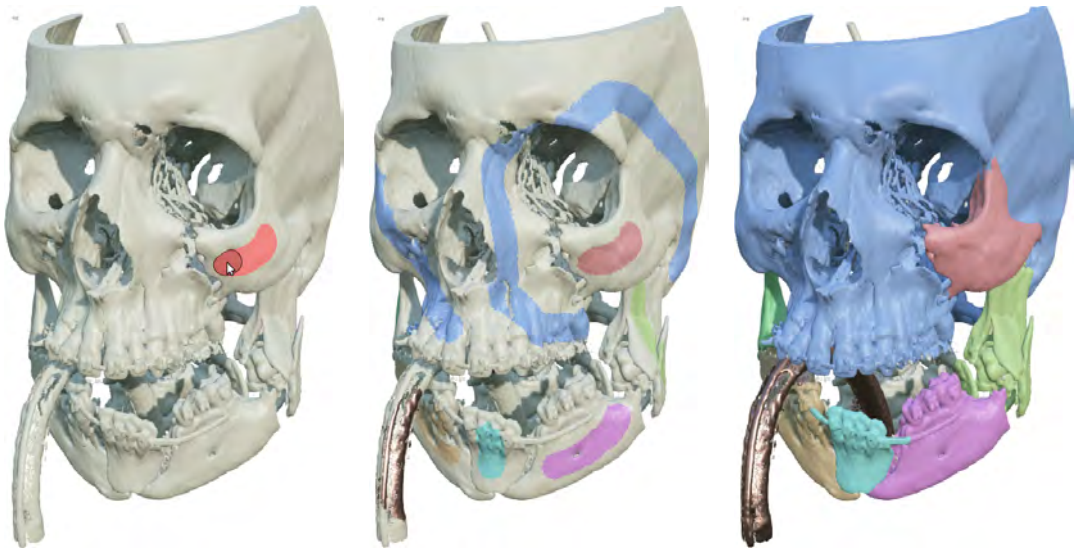
Figure 3: Deferred isosurface shading pipeline: (a) ray-start positions; (b) first-hit texture; (c) surface normals; (d) shadow map derived from an additional first-hit texture rendered from a directional light source's point of view; (e) local illumination; (f) local illumination with shadows.

algorithm since it is robust to noise and weak boundaries, extends easily to multi-label (K-way) segmentation, and does not suffer from the small-cut problem of graph cuts. The main drawback and limitation of random walks is its high computational and memory cost (which, to be fair, is also a problem for graph cuts). For interactive multi-label segmentation of volume images, this has traditionally limited the maximum volume size to around 256^3, which is smaller than the CT volumes normally encountered in CMF planning. Our random walks implementation overcomes this limitation by only operating on bone voxels.

We construct a weighted graph $G = (V, E)$ from the collective bone segmentation and use the random walks algorithm to separate individual bones marked by the user. Figure 4 illustrates the segmentation process. For every bone voxel, the random walks algorithm calculates the probability that a random walker starting at the voxel will reach a particular seed label. A crisp segmentation is obtained by, for each bone voxel, selecting the label with the highest probability value. The vertices $v \in V$ in the graph represent the bone voxels and the edges $e \in E$ represent the connections between adjacent bone voxels in a 6-connected neighborhood. The number of neighbors can vary from zero to six. Each edge e_{ij} between two neighbor vertices v_i and v_j is assigned a gradient magnitude-based weight w_{ij} [12] defined as

$$w_{ij} = \exp(-\beta(g_i - g_j)^2) + \varepsilon, \qquad (1)$$

where g_i and g_j are the intensities of v_i and v_j in the underlying grayscale image, and β is a parameter that determines the influence of the gradient magnitude. We add a small positive constant ε (set to 0.01 in our implementation) to ensure that v_i and v_j are connected, i.e., $w_{ij} > 0$. Increasing the value of β makes the random walkers less prone to traverse edges with high gradient magnitude. Empirically, we have found $\beta = 3000$ to work well for bone separation; however, the exact choice of β is not critical and we have used values in the range 2000-4000 with similar results.



(a) (b) (c)

Figure 4: 3D texture painting interface for interactive random walks segmentation: (a) 3D brush used for painting seeds directly on the bone surfaces; (b) marked bones; (c) bone separation obtained with random walks.

As proposed in [12], we represent the weighted graph and the seed nodes as a sparse linear system and use an iterative solver (see Section 2.5) to approximate the solution for each label. By constructing the graph from the bone voxels in the collective segmentation, rather than from the full image, we simplify the random walks segmentation task from separation of multiple tissue types to bone separation. Moreover, we reduce the memory and computational cost substantially (by $\sim 90\%$ in our test cases). The head CT volumes encountered in CMF planning typically contain between 3 and 8 million bone voxels, which is a small fraction, $\sim 10\%$, of the total number of voxels. Combined with fast iterative solvers, this enables rapid update of the segmentation for volumes as large as 512^3.

A problem with constructing graphs from collective bone segmentations is that the sparse matrix A in the linear system becomes singular if some of the bone voxels are isolated (which, due to noise, is often the case.) This prevents the iterative solver from converging to a stable solution. The problem does not occur for graphs constructed from full images, where every voxel has at least one neighbor. To remove the singularity, we simply add a small constant weight $\kappa = 0.001$ to the diagonal elements in A. The value of κ is set smaller than ε to not interfere with the gradient weighting.

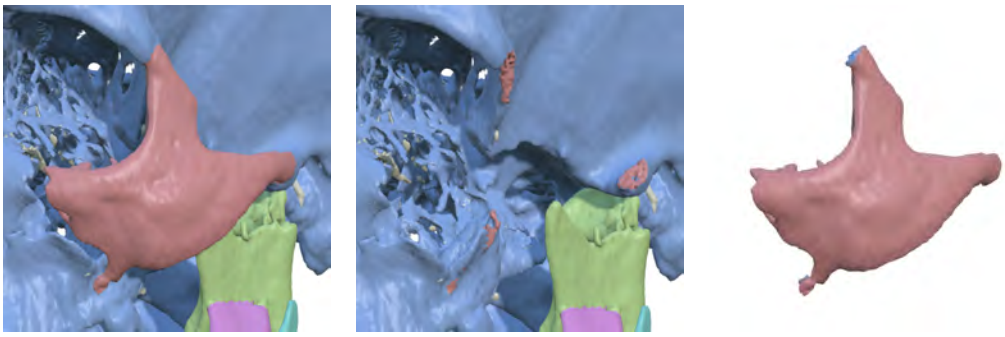
2.5 Iterative Solvers

We compute the random walks probability values iteratively using the Jacobi preconditioned conjugate gradient (CG) [28] method. The CG solver consists of dense vector operations and a sparse matrix-vector multiplication (SpMV), where SpMV is the most expensive operation. Although the Jacobi preconditioner improves the convergence rate, we found a single-threaded CPU implementation to be too slow for our problem sizes. Hence, to enable an interactive workflow, we followed the suggestion in [12] and implemented multi-threaded and GPU-accelerated versions of the solver. The multi-threaded solver was implemented in OpenMP and uses the compressed sparse row (CSR) matrix format for SpMV. The GPU-accelerated solver was implemented in OpenCL and supports two sparse matrix formats: CSR and ELLPACK [3]. ELLPACK has a slightly higher memory footprint than CSR, but enables coalesced memory access when executing the SpMV kernel on GPUs, which usually leads to better performance [3]. Our OpenCL SpMV kernels are based on the CUDA implementations in [3]. A benchmark of the implemented solvers is presented in Section 3.

2.6 Segmentation Editing

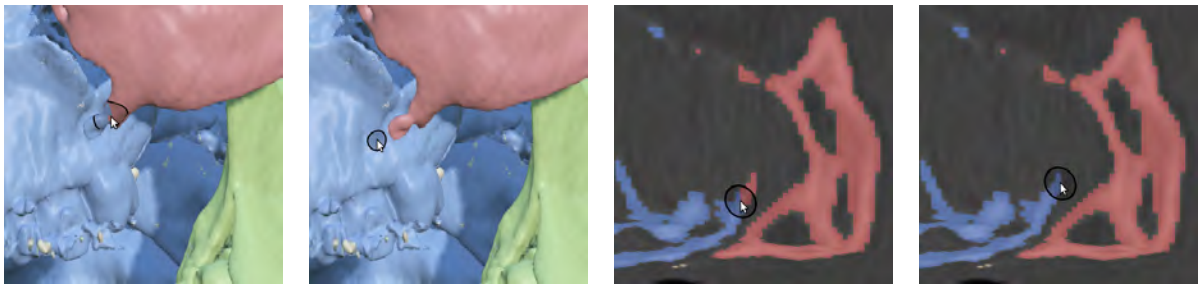
The user can edit the initial random walks segmentation by painting additional seeds on the bone surfaces or individual CT slices and running the iterative solver again. To enable rapid update of the result, the previous solution is used as starting guess [12]. Visual inspection is supported by volume clipping (Figure 5). The editing process can be repeated until an acceptable segmentation result has been obtained.

Further refinement of the segmentation can be achieved with a dedicated 3D editing tool (Figure 6), which updates a local region of the segmentation in real-time and allows a selected label to grow and compete with other labels. The tool is represented as a spherical brush and affects only voxels within the brush radius r. A voxel p_i marked with the active label will transfer its label to an adjacent voxel p_j in a 26-neighborhood if the editing



(a) (b) (c)

Figure 5: To support visual inspection and editing of bone fragments and contact surfaces, a segmented region (a) can be hidden (b) or exposed (c) via volume clipping . The clipping is performed by temporarily setting the grayscale value of the segmented region to $t_{bone} - 1$ and updating the grayscale 3D texture.



(a) (b) (c) (d)

Figure 6: Segmentation editing performed with the local 3D editing tool.

weight function W_{ij} exceeds a given threshold. W_{ij} is defined as a weighted sum of the active label ratio, the gradient, and the Euclidean distance to the brush center.

2.7 Implementation Details

We implemented the segmentation system in Python, using OpenGL and GLSL for the rendering, PySide for the graphical user interface, and Cython and PyOpenCL for the image and graph processing.

3 CASE STUDY

To demonstrate the efficiency of our tool, we asked two non-medical test users to perform interactive segmentations of the facial skeleton in CT scans of three complex CMF cases. The first user, who had prior experience of manual bone segmentation and virtual surgery planning, was a novice on the system and received a 15 minutes training session before the segmentations started, whereas the second user (the main author) was an expert on the system. The CT scans were obtained as anonymized DICOM files. Further details about the datasets are provided in Table 1. Figures 7a–7c show the collective bone segmentations obtained by thresholding. Bone separation was carried out in three stages:

1. Initial random walks segmentation of marked bones.
2. Interactive coarse editing of the segmentation result by running random walks multiple times with additional seed strokes as input.
3. Fine-scale editing with the local 3D editing tool.

We measured the computational time and the interaction time required for each stage and asked the users to save the segmentation result obtained in each stage. Additionally, one of the users segmented case 1 manually in the ITK-SNAP [30] software to generate a reference segmentation for accuracy assessment. The manual segmentation took ∼5 hours to perform and was inspected and validated by a CMF surgeon.

To assess segmentation accuracy and precision, we computed the Dice similarity coefficient

$$DSC = \frac{2|A \cap B|}{|A| + |B|}. \tag{2}$$

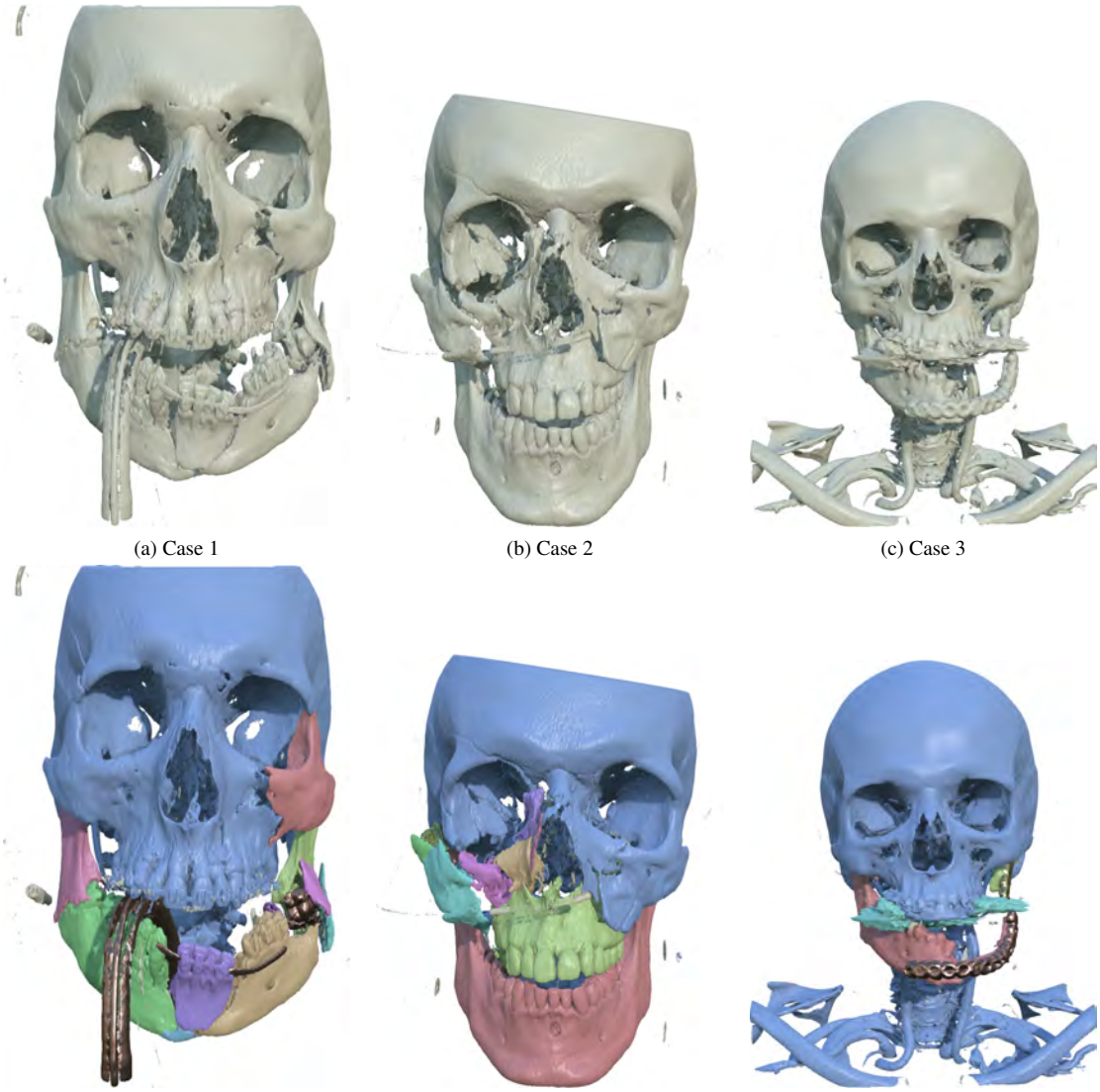
DSC measures the spatial overlap between two multi-label segmentations A and B and has the range $[0,1]$, where 0 represents no overlap and 1 represents complete overlap.

The interactive segmentations (Figures 7d–7f) took on average 14 minutes to perform. As shown in Figure 8, most of the time was spent in the local editing stage (stage 3). DSC between the final interactive case 1 segmentations and the manual reference segmentation was 0.97782 (User 1) and 0.97784 (User 2), indicating overall high spatial overlap. The inter-user precision (Table 2) was also high and improved with editing.

Figure 9 shows a benchmark of the implemented CG solvers. The bars show the execution times (in sec-

Case	Region	Description	#Labels	Dimensions	Threshold	#Bone voxels
1	Head	Multiple fractures	15	$512 \times 512 \times 337$	260	4426530
2	Head	Multiple fractures	12	$512 \times 512 \times 301$	300	4769742
3	Head	Tumor	6	$230 \times 512 \times 512$	300	2787469

Table 1: Details about the CT images used in the case study.



(a) Case 1 (b) Case 2 (c) Case 3

(d) Case 1 (e) Case 2 (f) Case 3

Figure 7: Top row: Collective bone segmentations. Bottom row: Separated bones.

onds) for computing an initial random walks solution on a graph with 4.6M bone voxels and 15 labels. The fastest GPU-based implementation had an average execution time of 0.4 seconds per label, which is a 14× speedup compared with the single-threaded CPU implementation and a 7× speedup compared with the multi-threaded CPU implementation.

4 DISCUSSION

Overall, we found the performance of the bone separation tool to be acceptable for surgery planning. Minor differences between segmentations generated by different users and between interactive and manual segmentations were expected due to the complex boundaries of the bone structures and the interactive editing.

Local editing (stage 3) is the most time-consuming part of the segmentation. The editing tool is of great aid for cleaning up the random walks segmentation and refining contact surfaces between separated bones or bone fragments, but will sometimes grow the active label too far or produce isolated voxels. Further modifications of the weight function could prevent this. Using connected component analysis for removing small isolated components in the segmentation could also be useful.

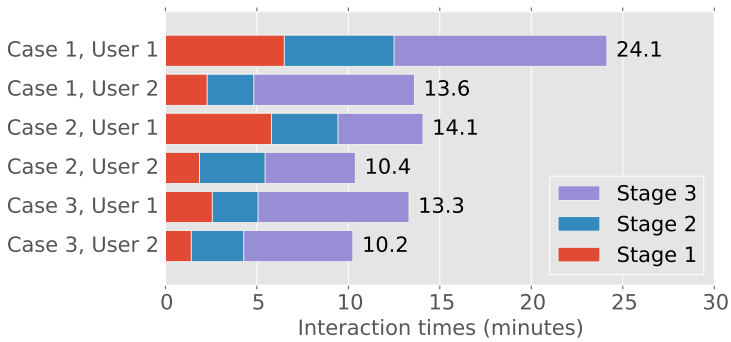


Figure 8: Interaction times (in minutes) for the two users.

Case	DSC		
	Stage 1	Stage 2	Stage 3
1	0.9199	0.9955	0.9971
2	0.9533	0.9968	0.9971
3	0.9832	0.99	0.9915

Table 2: Inter-user precision for the interactive segmentations.

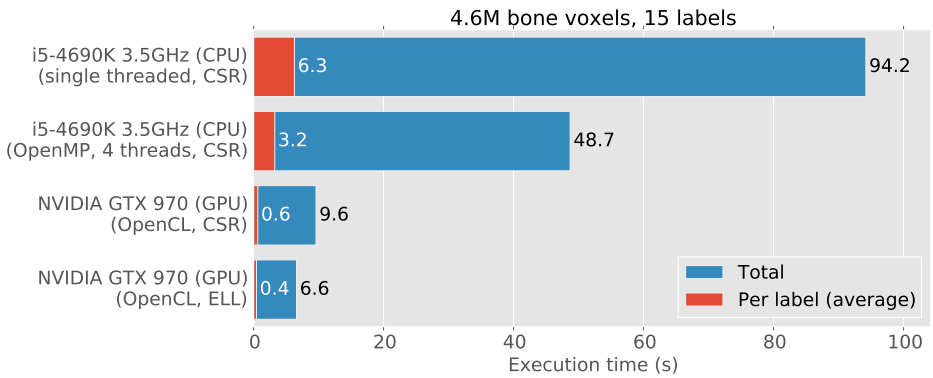


Figure 9: Benchmark of the CPU- and GPU-based Jacobi preconditioned CG solvers. The graph shows the timings (in seconds) for computing the initial random walks solution on a graph with 4.6M bone voxels and 15 labels. The number of iterations per label ranged from 45 to 136 (mean 83). Solver tolerance was set to $3 \cdot 10^{-3}$.

A limitation of our current approach is that the initial thresholding segmentation either tend to exclude thin or low-density bone structures or include noise and soft tissue. However, with minor modifications, the system should be able to display and process collective bone segmentations generated with other segmentation techniques. Postprocessing could potentially fill in holes.

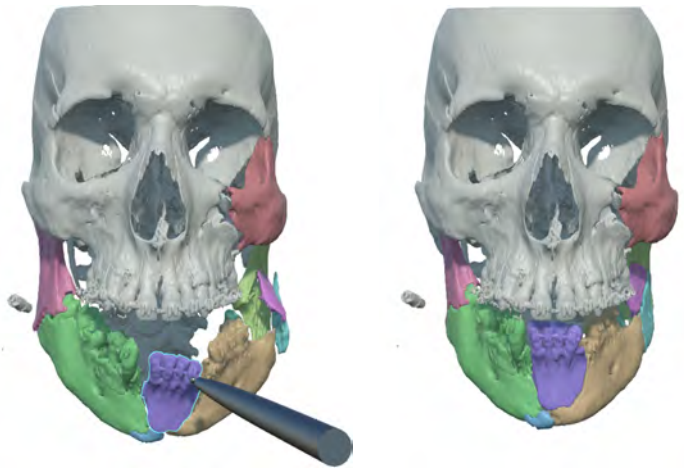
5 CONCLUSION AND FUTURE WORK

In this paper, we have presented an efficient 3D texture painting tool for segmenting individual bone structures in 3D CT images. This type of segmentation is crucial for virtual CMF surgery planning [25], and can take several hours to perform with conventional manual segmentation approaches. Our tool can produce an accurate segmentation in a few minutes, thereby removing a major bottleneck in the planning procedure. The resulting segmentation can, as demonstrated in Figure 10, be used as input for virtual assembly [25]. Our tool is not limited to CMF planning, but can also be used for orthopedic applications or fossil data (Figure 11).

Next, we will focus on improving the efficiency of the local editing tool. We will also investigate if the accuracy of the random walks segmentation can be improved by combining the gradient-based weight function with other weight functions based on, for example, bone sheetness measure [7] or local edge density [22]. Finally, we will apply our segmentation tool on a larger set of CT images and perform a more extensive evaluation of the precision, accuracy, and efficiency.

6 ACKNOWLEDGMENTS

Our thanks go to Fredrik Nysjö (Centre for Image Analysis, Uppsala University) for performing the virtual assembly and generating reference segmentations for the case study. The head and wrist CT scans were provided by Uppsala University Hospital. The pelvis and



(a) (b)

Figure 10: Haptic-assisted virtual assembly of one of the segmented cases, performed with the HASP [25] system.

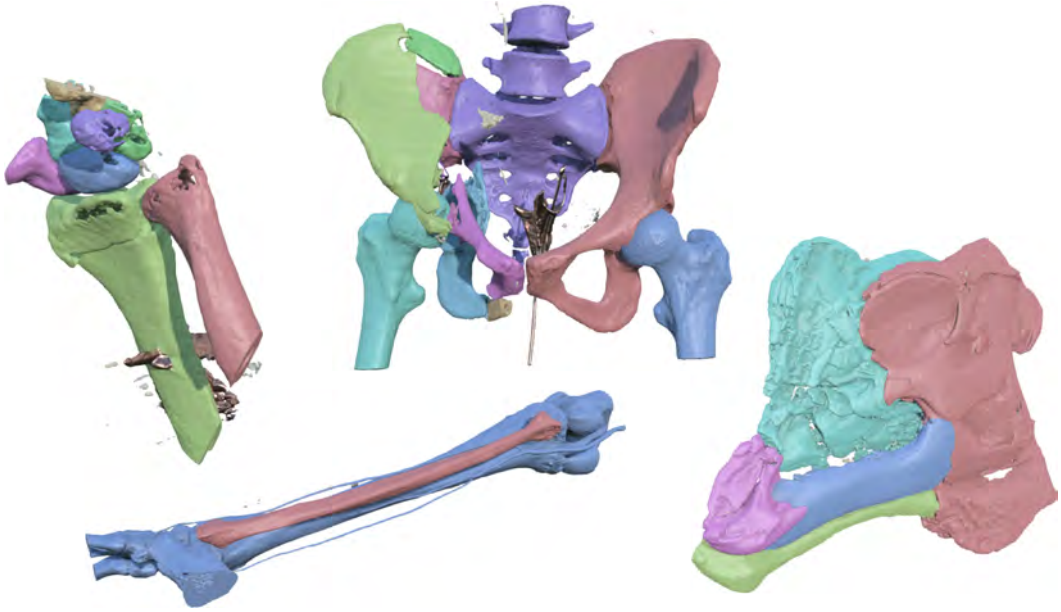


Figure 11: Our tool is not limited to head and neck CT scans; it can be used for rapid segmentation of individual bone structures in other regions such as the wrist, lower limbs, and pelvis. Another potential application (shown in the right image) is segmentation of fossils in μCT scans. Total segmentation time for these four cases was < 1 h.

fibula scans are courtesy of the OsiriX DICOM repository (http://www.osirix-viewer.com/datasets/), and the fossil μCT scan is courtesy of [8].

7 REFERENCES

[1] T. Akenine-Möller, E. Haines, and N. Hoffman. *Real-Time Rendering 3rd Edition*. A. K. Peters, Ltd., Natick, MA, USA, 2008.

[2] T. Alathari, M. Nixon, and M. Bah. Femur Bone Segmentation Using a Pressure Analogy. In *22nd International Conference on Pattern Recognition (ICPR 2014)*, pages 972–977, 2014.

[3] N. Bell and M. Garland. Implementing Sparse Matrix-vector Multiplication on Throughput-oriented Processors. In *Conference on High Performance Computing Networking, Storage and Analysis*, SC '09, pages 1–11. ACM, 2009.

[4] Y. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in ND images. In *8th IEEE International Conference on Computer Vision (ICCV 2001)*, volume 1, pages 105–112, 2001.

[5] K. Bürger, J. Krüger, and R. Westermann. Direct volume editing. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1388–1395, 2008.

[6] A. Criminisi, T. Sharp, and A. Blake. Geos: Geodesic image segmentation. In *European Conference on Computer Vision (ECCV)*, volume 5302 of *LNCS*, pages 99–112. Springer, 2008.

[7] M. Descoteaux, M. Audette, K. Chinzei, and K. Siddiqi. Bone enhancement filtering: application to sinus bone segmentation and simulation of pituitary surgery. *Computer Aided Surgery*, 11(5):247–255, 2006.

[8] A. Farke and R. M. Alf Museum of Paleontolog. CT scan of left half of skull of Parasaurolophus sp. (Hadrosauridae: Dinosauria). 2013.

[9] J. Fornaro, G. Székely, and M. Harders. Semi-automatic Segmentation of Fractured Pelvic Bones for Surgical Planning. In *Biomedical Simulation*, volume 5958 of *LNCS*, pages 82–89. Springer Berlin Heidelberg, 2010.

[10] P. Fürnstahl, T. Fuchs, A. Schweizer, L. Nagy, G. Székely, and M. Harders. Automatic and robust forearm segmentation using graph cuts. In *5th IEEE International Symposium on Biomedical Imaging (ISBI 2008).*, pages 77–80, May 2008.

[11] P. Fürnstahl, G. Székely, C. Gerber, J. Hodler, J. G. Snedeker, and M. Harders. Computer assisted reconstruction of complex proximal humerus fractures for preoperative planning. *Medical Image Analysis*, 16(3):704–720, 2012.

[12] L. Grady. Random Walks for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1768–1783, 2006.

[13] M. Hadwiger, P. Ljung, C. R. Salama, and T. Ropinski. Advanced Illumination Techniques for GPU Volume Raycasting. In *ACM SIGGRAPH ASIA 2008 Courses*, SIGGRAPH Asia '08, pages 1–166. ACM, 2008.

[14] M. Hadwiger, C. Sigg, H. Scharsach, K. Bühler, and M. Gross. Real-Time Ray-Casting and Advanced Shading of Discrete Isosurfaces. *Computer Graphics Forum*, 24(3):303–312, 2005.

[15] H. K. Hahn and H.-O. Peitgen. IWT-interactive watershed transform: a hierarchical method for efficient interactive and automated segmentation of multidimensional gray-scale images. In *Medical Imaging 2003*, pages 643–653. SPIE, 2003.

[16] G. Hamarneh, J. Yang, C. McIntosh, and M. Langille. 3D live-wire-based semi-automatic segmentation of medical images. In *Medical Imaging 2005*, pages 1597–1603. SPIE, 2005.

[17] P. Hanrahan and P. Haeberli. Direct WYSIWYG Painting and Texturing on 3D Shapes. *ACM SIGGRAPH Computer Graphics*, 24(4):215–223, 1990.

[18] M. Krcah, G. Székely, and R. Blanc. Fully automatic and fast segmentation of the femur bone from 3D-CT images with no shape prior. In *IEEE International Symposium on Biomedical Imaging*, pages 2087–2090. IEEE, 2011.

[19] J. Krüger and R. Westermann. Acceleration Techniques for GPU-based Volume Rendering. In *14th IEEE Visualization 2003 (VIS'03)*, pages 287–292, 2003.

[20] Y.-K. Lai, S.-M. Hu, R. R. Martin, and P. L. Rosin. Rapid and Effective Segmentation of 3D Models Using Random Walks. *Computer Aided Geometric Design*, 26(6):665–679, 2009.

[21] H. Landis. Production-ready global illumination. *SIGGRAPH Course Notes*, 16(2002):11, 2002.

[22] L. Liu, D. Raber, D. Nopachai, P. Commean, D. Sinacore, F. Prior, R. Pless, and T. Ju. Interactive Separation of Segmented Bones in CT Volumes Using Graph Cut. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI 2008)*, volume 5241 of *LNCS*, pages 296–304. Springer Berlin Heidelberg, 2008.

[23] C. S. Mendoza, N. Safdar, K. Okada, E. Myers, G. F. Rogers, and M. G. Linguraru. Personalized assessment of craniosynostosis via statistical shape modeling. *Medical Image Analysis*, 18(4):635–646, 2014.

[24] A. Neubauer, K. Bühler, R. Wegenkittl, A. Rauchberger, and M. Rieger. Advanced virtual corrective osteotomy. *International Congress Series*, 1281(0):684–689, 2005.

[25] P. Olsson, F. Nysjö, J.-M. Hirsch, and I. B. Carlbom. A haptics-assisted cranio-maxillofacial surgery planning system for restoring skeletal anatomy in complex trauma cases. *International Journal of Computer Assisted Radiology and Surgery*, 8(6):887–894, 2013.

[26] W. T. Reeves, D. H. Salesin, and R. L. Cook. Rendering antialiased shadows with depth maps. *ACM SIGGRAPH Computer Graphics*, 21(4):283–291, 1987.

[27] S. M. Roser and et al. The accuracy of virtual surgical planning in free fibula mandibular reconstruction: comparison of planned and final results. *Journal of Oral and Maxillofacial Surgery*, 68(11):2824–2832, 2010.

[28] J. R. Shewchuk. *An introduction to the conjugate gradient method without the agonizing pain*. Carnegie-Mellon University. Department of Computer Science, 1994.

[29] The Foundry. MARI. `http://www.thefoundry.co.uk/products/mari/`, 2015. Accessed on February 21, 2015.

[30] P. A. Yushkevich, J. Piven, H. Cody Hazlett, R. Gimpel Smith, S. Ho, J. C. Gee, and G. Gerig. User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. *Neuroimage*, 31(3):1116–1128, 2006.