ISSN 1213-6972 *Volume 26, Number 2, 2018*

Journal of WSCG

An international journal of algorithms, data structures and techniques for computer graphics and visualization, surface meshing and modeling, global illumination, computer vision, image processing and pattern recognition, computational geometry, visual human interaction and virtual reality, animation, multimedia systems and applications in parallel, distributed and mobile environment.

Vaclav Skala – Union Agency

Journal of WSCG

Editor-in-Chief: Vaclav Skala
c/o University of West Bohemia
Faculty of Applied Sciences
Univerzitni 8
CZ 306 14 Plzen
Czech Republic
http://www.VaclavSkala.eu

Managing Editor: Vaclav Skala

Printed and Published by:
Vaclav Skala - Union Agency
Na Mazinach 9
CZ 322 00 Plzen
Czech Republic

Published in cooperation with the University of West Bohemia
Univerzitní 8, 306 14 Pilsen, Czech Republic

Hardcopy: ***ISSN 1213 – 6972***
CD ROM: ***ISSN 1213 – 6980***
On-line: ***ISSN 1213 – 6964***

ISSN 1213-6972 *Volume 26, Number 2, 2018*

Journal of WSCG

An international journal of algorithms, data structures and techniques for computer graphics and visualization, surface meshing and modeling, global illumination, computer vision, image processing and pattern recognition, computational geometry, visual human interaction and virtual reality, animation, multimedia systems and applications in parallel, distributed and mobile environment.

Vaclav Skala – Union Agency

Journal of WSCG

Editor-in-Chief: Vaclav Skala
c/o University of West Bohemia
Faculty of Applied Sciences
Univerzitni 8
CZ 306 14 Plzen
Czech Republic
http://www.VaclavSkala.eu

Managing Editor: Vaclav Skala

Printed and Published by:
Vaclav Skala - Union Agency
Na Mazinach 9
CZ 322 00 Plzen
Czech Republic

Published in cooperation with the University of West Bohemia
Univerzitní 8, 306 14 Pilsen, Czech Republic

Hardcopy: ***ISSN 1213 – 6972***
CD ROM: ***ISSN 1213 – 6980***
On-line: ***ISSN 1213 – 6964***

Journal of WSCG

Editor-in-Chief

Vaclav Skala

c/o University of West Bohemia
Faculty of Applied Sciences
Department of Computer Science and Engineering
Univerzitni 8, CZ 306 14 Plzen, Czech Republic
http://www.VaclavSkala.eu

Journal of WSCG URLs: http://www.wscg.eu or http://wscg.zcu.cz/jwscg

Editorial Board

Journal of WSCG

Board of Reviewers
2018

Aburumman,N. (France)
Assarsson,U. (Sweden)
Ayala,D. (Spain)
Azari,B. (Germany)
Benes,B. (United States)
Benger,W. (United States)
Bouatouch,K. (France)
Bourke,P. (Australia)
Carmo,M. (Portugal)
Carvalho,M. (Brazil)
Daniel,M. (France)
de Geus,K. (Brazil)
De Martino,J. (Brazil)
de Souza Paiva,J. (Brazil)
Dingliana,J. (Ireland)
Durikovic,R. (Slovakia)
Feito,F. (Spain)
Feng,J. (China)
Ferguson,S. (United Kingdom)
Galo,M. (Brazil)
Galo,M. (Brazil)
Garcia Hernandez,R. (Germany)
Garcia-Alonso,A. (Spain)
Gavrilova,M. (Canada)
Gdawiec,K. (Poland)
Giannini,F. (Italy)
Goncalves,A. (Portugal)
Gudukbay,U. (Turkey)
Hernandez,B. (United States)
Horain,P. (France)
Charalambous,P. (Cyprus)
Juan,M. (Spain)
Kanai,T. (Japan)
Klosowski,J. (United States)
Kurt,M. (Turkey)
Lee,J. (United States)
Lisowska,A. (Poland)
Lobachev,O. (Germany)
Luo,S. (Ireland)
Marques,R. (Spain)
MASTMEYER,A. (Germany)
Metodiev,N. (United States)
Molla,R. (Spain)
Montrucchio,B. (Italy)
Muller,H. (Germany)
Oliveira,J. (Portugal)
Oyarzun Laura,C. (Germany)
Papaioannou,G. (Greece)
Patow,G. (Spain)
Pedrini,H. (Brazil)
Peytavie,A. (France)
Puig,A. (Spain)
Ramires Fernandes,A. (Portugal)

Renaud,c. (France)
Ribeiro,R. (Portugal)
Richardson,J. (United States)
Rodrigues,J. (Portugal)
Rojas-Sola,J. (Spain)
Sanna,A. (Italy)
Santos,L. (Portugal)
Segura,R. (Spain)
Skala,V. (Czech Republic)
Sousa,A. (Portugal)
Subsol,G. (France)
Szecsi,L. (Hungary)
Tavares,J. (Portugal)
Teschner,M. (Germany)
Thalmann,D. (Switzerland)
Todt,E. (Brazil)
Tokuta,A. (United States)
Trapp,M. (Germany)
Vanderhaeghe,D. (France)
Vidal,V. (France)
Vierjahn,T. (Germany)
Wuensche,B. (New Zealand)
Wuethrich,C. (Germany)
Xu,K. (China)
Yin,Y. (United States)
Yoshizawa,S. (Japan)
Zwettler,G. (Austria)

Journal of WSCG
Vol.20, No.2

Contents

Yan,L, Gauthier,L.: Scalable Light Field Disparity Estimation with Occlusion Detection 66

Discher,S., Masopust,L., Schulz,S., Richter,R., Döllner,J.: A Point-Based and Image-Based Multi-Pass Rendering Technique for Visualizing Massive 3D Point Clouds in VR Environments 76

Eaksarayut,W., Tunwattanapong,B., Sitthi-amorn,P., Chentanez,N.: Mesh-based Multi-view Normal Integration with Energy Minimization Using Surface Reflectance Properties 85

Mylo,M., Klein,R.: Linear Subspaces of the Appearance Space 94

Brahimi,S., Ben Aoun,N., Ben Amar,C., Benoit,A., Lambert,P.: Multiscale Fully Convolutional DenseNet for Semantic Segmentation 104

Castillo,S., Cunningham,D.W., Winger,C., Breuß,M.: Morphological Amoeba-based Patches for Exemplar-Based Inpainting 112

Del Gallego,N., Ilao,J.: Improving Multiple-Image Super-Resolution for Mobile Devices through Image Alignment Selection 122

ISSN 1213-6972 Journal of WSCG http://www.wscg.eu/ Vol.26, 2018, No.2

Scalable Light Field Disparity Estimation with Occlusion Detection

Yan Li
LISA department
Université Libre de Bruxelles
Brussels, Belgium
yali@ulb.ac.be

Gauthier Lafruit
LISA department
Université Libre de Bruxelles
Brussels, Belgium
gauthier.lafruit@ulb.ac.be

ABSTRACT

An occlusion-aware framework is proposed to robustly estimate the disparities of light field images. It is mainly realized by leveraging multiple edge cues to occlusion detection and then integrate it with local costs into an energy function. To check the performance, the quantitative and/or qualitative evaluations are performed on both synthetic and natural light field datasets. It demonstrates that the proposed framework is robust to the density and disparity range of the light field, advancing the state-of-the-art light field disparity estimation frameworks on aspect of accuracies.

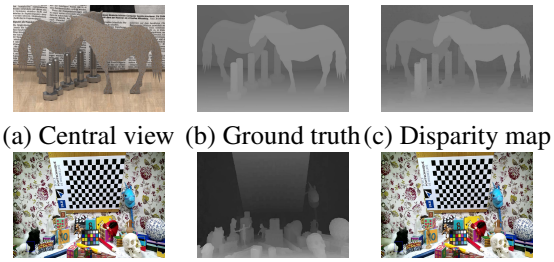
Keywords

Light Field, Disparity/Depth Estimation, Occlusion Detection, Global Optimization.

1 INTRODUCTION

Contrary to a traditional 2D image, the light field records not only the radiance but also the direction of a light ray. This richer information of the light field motivates a large range of computer vision and graphics applications, including disparity/depth estimation [Wanner and Goldluecke, 2012, Kim et al., 2013, Chen et al., 2014, Jeon et al., 2015, Wang et al., 2015, Zhang et al., 2016, Zhu et al., 2017], digital refocusing [Ng et al., 2005] and super-resolution [Wanner and Goldluecke, 2014], etc. In this work, our focus is put on disparity/depth estimation, which is employed as a module of view synthesis for virtual reality (VR) [Huang et al., 2017, MPEG-I, 2017].

Disparity estimation, is a long-term challenging issue in computer vision, which finds correspondences from stereo image pairs. The well-generated disparity maps from this task could bring benefits to various applications, such as view synthesis [Stankiewicz et al., 2013], superpixel segmentation [Stutz et al., 2018], semantic segmentation [Zhang et al., 2010], etc. A common solution for disparity estimation is to employ two views, namely stereo matching [Scharstein and Szeliski, 2002]. Since more viewpoints are available in the light field (Fig. 2), more accurate disparity estimations are possible than in stereo matching.



(a) Central view (b) Ground truth (c) Disparity map

(d) Central view (e) Depth map (f) Synthesized map

Figure 1: Disparity/Depth estimation results on light field images. The top shows the proposed disparity map of the central view in a dense light field; The bottom shows the proposed depth map of the central view and its corresponding synthesized/virtual map in a sparse and large disparity range of the light field, in which the synthesized map is obtained by view synthesis using two neighboring views and their depth maps.

Nowadays, the state-of-art light field disparity estimation references achieve a significantly high accuracy when the disparity range between sub-aperture images is narrow and the light field is densely sampled. However, we observe that the accuracy still remains an issue when the disparity range between sub-aperture images is larger and the light field is sparser. Moreover, this is non-trivial in virtual view rendering of MPEG-I activities [MPEG-I, 2017] in which a sparser and larger disparity range of the light field is being used.

To cope with this issue, a scalable framework for light field disparity estimation is proposed in the paper. More specifically, the kernel density estimation and

https://doi.org/10.24132/JWSCG.2018.26.2.1

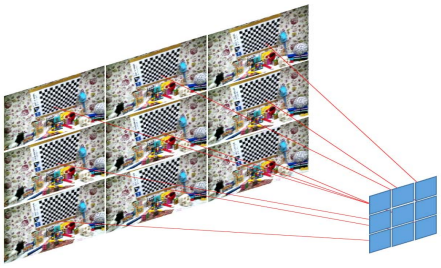


Figure 2: Light field images, also called sub-aperture images, are captured from an equally spaced 2D camera array.

size-adaptive window filter are introduced to locally estimate disparities in which an adaptive size is considered not to be sensitive to the disparity range (Sec 3.2). Since there are more ambiguities at occlusion areas, an occlusion handling method, i.e., occlusion detection and score-volume recomputation, are proposed (Sec 3.3), followed by using an occlusion-aware optimization to improve disparity continuity and enforce global consistency (Sec 3.4). The experimental results show that the proposed framework produces a high accuracy of disparity/depth maps in multiple densities and disparity ranges of light fields, see Fig. 1.

The contributions of our work are summarized below:

1. The proposed framework significantly advances state-of-the-art reference work on both synthetic and natural light field datasets in disparity/depth accuracy.
2. The accuracy of disparity maps from the proposed method is more robust to the density and disparity range of the light field, achieving at least 14.9%, 19.6%, 29.3% Root Mean Square Error (RMSE) gains on average in 9x9, 5x5 and 3x3 synthetic light fields respectively when compared to the state-of-art works.
3. An occlusion handling technique using multiple edge cues is put forward, which always benefits disparity estimations at occlusion regions.

2 RELATED WORK

Since the light field has the redundant information, some works formulate the problem of disparity estimation as the slope calculation of the Epipolar Plane Image (EPI) without explicit occlusion detection. [Wanner and Goldluecke, 2012] introduce a structure tensor technique to light field disparity estimation, and use it to compute the slope of EPI, followed by the integration into a variational-based energy function. However, its accuracy is confined to a narrow disparity range. [Kim et al., 2013] put forward another EPI-based framework using a fine-to-coarse strategy. Since it relies on a pixel to match the corresponding EPI-pixel, it seems robust at occlusion regions but the light field has to be guaranteed densely sampled. [Zhang et al., 2016] propose a spinning parallelogram operator onto both horizontal and vertical EPI-lines, preserving sharp disparity edges. [Jeon et al., 2015] present a disparity estimation framework in sub-pixel accuracy. However, both methods are subject to insufficient disparity quality at textureless regions, though they are not much influenced by the density and disparity range of light field.

Some other works turn to the Surface Camera (SCam)-based strategy to perform disparity estimation with explicit occlusion detection. In fact, the occlusion is a tough issue in matching correspondences. Multi-view stereo matching makes an early effort to this occlusion issue. [Kolmogorov and Zabih, 2002] describe a graph cut framework in which the visibility term is formulated into an energy function. [Wei and Quan, 2005] propose an asymmetric model to overcome occlusions in an efficient way. However, the heavy occlusion still remains an issue, even with a large number of views. To handle (heavy) occlusions, some recent light field-based methods are proposed [Chen et al., 2014, Wang et al., 2015, Zhu et al., 2017]. [Chen et al., 2014] introduce a bilateral consistency metric to predict the occlusion at SCam reliably. [Wang et al., 2015] describe an angular patch occlusion model, i.e., a single-occluder model, in which edge detection is required to obtain the occlusion boundary. When a multiple-occluder appears, it cannot work well because the single-occluder assumption does not hold. To overcome this drawback, [Zhu et al., 2017] describe a multiple-occluder modeling and then adopt an un-occluded view selection and re-selection scheme. Since its accuracy relies on the occlusion boundary, a disparity edge map is combined with an edge map to improve occlusion boundary detections. However, these works are somewhat restricted to dense light fields.

In contrast, the proposed framework, which is modeled by filter-based kernel density estimations with a separate occlusion-aware optimization technique, is not limited to the dense or narrow disparity range of the light field. The experimental results demonstrate that our method achieves a significantly high accuracy in multiple densities and disparity ranges of light fields, advancing the state-of-the-art performance.

3 APPROACH

Fig. 3 shows an overview of our approach. Taking a central view of light fields (Sec 3.1) for instance, the local disparity map (LDM) is initially produced from a winner-take-all strategy onto score-volume computations (Sec 3.2). Then a disparity edge map (DEM), canny edge map (CEM), superpixel edge map (SEM), occluded pixels map (OPM) are put into the occlusion handling site to extract an occlusion boundary map (OBM) and tweak the score of occluded pixels (Sec 3.3). With the aid of these occlusion detection results,

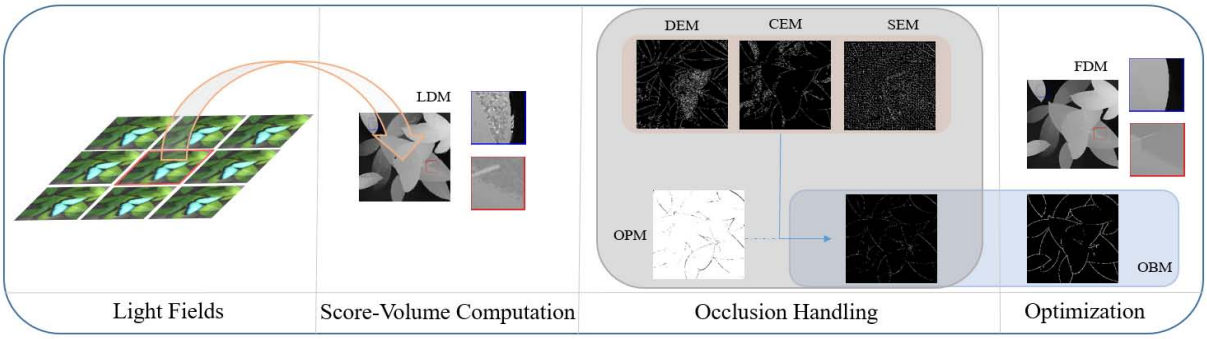


Figure 3: The proposed framework.

the final disparity map (FDM) is better generated under optimizations when compared with the LDM (Sec 3.4).

3.1 Light Fields

The light field, in the paper, is represented by two-plane parametrization (2PP) in which a camera plane is parametrized by the coordinate system (s,t) and the image plane (u,v). Then it could be simply seen as a collection of a plane of views with radiance values r in the RGB color space, described as $r = L(s,t,u,v)$, in which (s,t) represents a camera coordinate and (u,v) indicates a coordinate of a pixel on the image plane. The light field view, which is being estimated, is denoted by R_{s^*,t^*}. Then, according to this view, a radiance set $R_{s,t,u,v}(d)$ is easily built by assigning a hypothetical disparity d to light rays or pixels, as given in Eq. 1:

$$R_{s,t,u,v}(d) = \{L(s,t,u+d*(s^*-s),v+d*(t^*-t)) \mid s=1,2,...,M;t=1,2,...,N\} \quad (1)$$

where (M, N) denotes the angular resolution of the light field. The subscript (u,v) that corresponds to the pixel or light ray in a view is replaced with p in the following texts for simplicity.

3.2 Score-Volume Computation

Our score-volume computation is composed of two steps: 1) initially computing the score volume, 2) filtering the score volume. The (filtered) score volume indicates a 3D array (u,v,d) that stores the scores/probabilities of candidate disparities d for a pixel p in a light field view.

A kernel density function is employed to the initial score volume calculations, which is formulated as follows:

$$S_p(d) = \frac{1}{|\Omega|} \sum_{s,t \in \Omega} K_h\left(R_{s^*,t^*,p} - R_{s,t,p}(d)\right) \quad (2)$$

where $S_p(d)$ is the score of the pixel p of the being estimated view R_{s^*,t^*} at the candidate disparity d where the maximum value corresponds to the true disparity in volume, and Ω represents a number of valid views for score computations. $K_h(\cdot)$ corresponds to the Epanechnikov kernel that is given in Eq. 3 and h is its bandwidth parameter (= 0.02), which controls the accuracy of the density estimation. Actually, a higher value of h increases the accuracy and robustness to noise. However, it will lose fine details.

$$K_h(x) = \begin{cases} 1 - \left\|\frac{x}{h}\right\|^2 & \left\|\frac{x}{h}\right\| \le 1 \\ 0 & otherwise \end{cases} \quad (3)$$

Rather than the increase of h, a window-based filter, i.e., an edge-aware preserving filter [He et al., 2010], is introduced to filter out some noises, which is computed as follows:

$$\widetilde{S}_p(d) = \sum_q W_{pq} S_q(d) \quad (4)$$

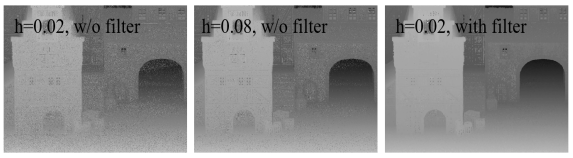
where $\widetilde{S}_p(d)$ is the filtered score of $S_p(d)$ and $S_q(d)$ is the score of the neighboring pixel q in a window. Since the filtering adopts an integral image based technique, it has a low complexity burden O(N). The weight of this filter is computed as below,

$$W_{pq} = \frac{1}{|\omega|^2} \sum_{k:(p,q)\in\omega_k} \left\{1 + \frac{(I_p - \mu_k)(I_q - \mu_k)}{\sigma_k{}^2 + \varepsilon}\right\} \quad (5)$$

where $W_{p,q}$ gives a higher weight to the pixel on the same side of the edge and a lower weight to the pixel on opposite sides of the edge in a window ω_k centered at the pixel k. The side length of this window ω_k is adaptive to the spatial resolution (w,h) of the light field, i.e., $max(\lfloor max(w,h)^2/(256*min(w,h))\rfloor, 3)$. I is a guided image, namely the light field view R_{s^*,t^*} that is being estimated; μ_k and σ_k are the mean and variance of the window ω_k in I respectively; ε is set to 0.01; $|\omega|$ is the number of the pixels in ω_k. The more effectiveness of this technique than the only increase of the h is shown in Fig. 4, clearly reducing the speckle noise.

3.3 Occlusion Handling

Assuming that the scene in light fields is lambertian, the scene point that is seen from different viewpoints shares



RMSE 0.4183 *RMSE* 0.2891 *RMSE* 0.1139

Figure 4: Compared with the increase of h, the edge-preserving filter demonstrates its higher ability (a lower RMSE) to remove the noises without losing fine details.

the same color, exhibiting the photo-consistency. However, this is not true for the point that is occluded. Some pixels from such a point in the score-volume computation step might be correctly estimated due to the edge-aware score volume computation. Nevertheless, the disparities of pixels at heavy occlusion regions still remain difficult to be well-estimated due to ambiguities. As a result, a pixel with a wrong disparity may be assigned a highest score. To address this issue, the occluded pixel detection (OPD), occlusion boundary detection (OBD) and score-volume recomputation (SVR) are proposed.

3.3.1 Occluded Pixel Detection

Some pixels disappear in parts of the views due to occlusions, breaking off the photo-consistency. Assuming that the scene is lambertian, a simple thresholding technique could be applied to detect these occluded pixels and obtain the occluded pixel map OPM, as given below,

$$C_p(d) = \frac{1}{|\Omega|}\sum_{\Omega}(1 - exp(-|R_{s^*,t^*,p} - R_{s,t,p}(d)|)) \quad (6)$$

where $C_p(d)$ indicates the occlusion confidence of the pixel p of the view R_{s^*,t^*} at the estimated disparity d. If the confidence of a pixel is larger than a specified threshold τ (= 0.05), it is masked as an occluded pixel $(OP = 1)$; otherwise it is unoccluded $(OP = 0)$.

3.3.2 Occlusion Boundary Detection

Occlusion boundary detection is a significant step for the occlusion handling as its accuracy makes differences for the following disparity re-estimation and occlusion-aware optimization. To guarantee its precision, multiple edge cues are proposed to precisely detect occlusion boundaries.

Firstly, a fact to be known is that there always exist edges between an occluder and an occluded region, which is ascribed to lighting changes in-between. Thus the following lemma is given.

Lemma I. An occlusion boundary set OB_s is a proper subset of an edge set EG_s.

The edge set is approximately constructed in our work for efficiency, i.e., a union of edge points and edge lines,

$$EG_s \simeq EG_{point} \cup EG_{region} \quad (7)$$

where EG_{point} denotes the edge points that are acquired by an edge detector, and EG_{region} indicates the edges from a region/superpixel detector [Stutz et al., 2018]. Note that the region size is set to a smaller value so as to be not much larger than the objects in the scene. Additionally, a small region used in a superpixel detector could boost the edge accuracy.

The occlusion boundaries that belong to the occlusion boundary set OB_s are taken from the approximated edge set. Firstly, for the view R_{s^*,t^*}, a disparity edge map DEM is computed from a relatively reliable local disparity map LDM using a canny edge detector [Canny, 1986], and an edge map EM is intersected by the canny edge map CEM and the superpixel edge map SEM. Then we calculate an intersection of DEM and EM to get an initial occlusion boundary map OBM^i. Furthermore, the disparity variance in a 10x10 window and the difference operator are computed as masks to update the difference between OBM^i and themselves in order to remove edge point outliers,

$$\begin{aligned} EM^u &= M_{disp} * (EM - OBM^i) \\ DEM^u &= M_{\nabla} * (DEM - OBM^i) \end{aligned} \quad (8)$$

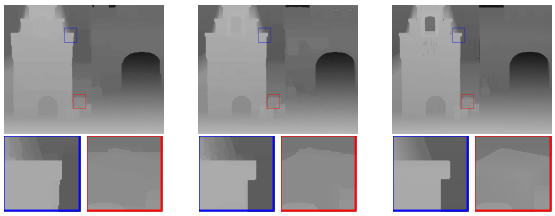
where M_{disp} and M_{∇} denote the disparity mask and the difference mask respectively. If the pixel has a disparity variance beyond a threshold φ that is adaptive to the disparity range, M_{disp} is assigned 1, otherwise 0. Similarly, if the pixel has a difference beyond a specified threshold $\nabla(= 0.05)$, M_{∇} is assigned as 1, otherwise 0. Finally, a union of multiple maps are used to produce the occlusion boundary map $OBM = OBM^i \cup DEM^u \cup EM^u$ with a high precision.

3.3.3 Score-Volume Recomputation

The score volume recomputation consists of two steps: 1) computing the disparity bound, 2) score-volume computation, targeting the improvement of the occluded pixel disparity estimation.

Disparity Bound The new upper bound ub and the lower bound lb in disparity are determined by the disparities of pixels in their neighborhood beforehand. The upper and lower bound are assigned to the maximum and minimum disparity of neighboring pixels respectively.

Score-Volume Computation The procedure in the previous score-volume computation is reused here, but there exist two differences. The first difference is that a disparity bound, i.e, a half-closed interval $[lb, ub)$, is utilized for computing the occluded pixel score $OccS_p(d)$ for the pixel p of the view R_{s^*,t^*} at a candidate disparity d. The second difference is that the visible views Ω_{vis} for photo-consistency are selected. More specifically, the relative location of the occluded pixel to the occlusion boundary from OBM (with rare negative occlusion boundaries) is used to simply select



w/o occ with occ Ground truth

Figure 5: Comparisons between without (w/o) and with occlusion detection results (occ) in the energy function. It demonstrates that the proposed occlusion-aware energy function contributes to a higher accuracy (a lower RMSE 0.099) without over-smoothing the sharp edges.

the visible views.
At the end of the occlusion handling flow, the occlusion boundary map with a high accuracy can be extracted and the score of the occluded pixel will be improved, which are beneficial to the following optimization step.

3.4 Optimization

Our disparity estimation is optimized by minimizing a Markov Random Field-based energy function, as given in Eq. 9.

$$E = \lambda * \sum_{p} E_{data}(p, d(p)) + \sum_{q \in N_p} E_{smooth}(p, q, d(p)) \tag{9}$$

where N_p is a 4-neighborhood of the pixel p of the view R_{s^*,t^*}, q represents one of the neighboring pixels and $d(p)$ denotes a disparity that is mapping to an integer. Herein λ (= 10) is introduced to balance the ratio of the data term and the smoothness term.
The data term in the energy function is built by weighting the score $\widetilde{S}$ and the occlusion score $OccS$,

$$E_{data}(p, d(p)) = \kappa - \alpha * \widetilde{S}_p(d) - (1-\alpha) * OccS_p(d) \tag{10}$$

where E_{data} measures the photo-consistency for the pixel p, α is a weighting coefficient (= 0.6) and κ is a large constant $(= 10)$.
The smoothness term is computed by a weighted neighboring function,

$$E_{smooth}(p, q, d(p)) = w_{p,q} * min(|d(p) - d(q)|, \Gamma) \tag{11}$$

$$w_{p,q} = exp(-\frac{||R_{s^*,t^*,p} - R_{s^*,t^*,q}||^2}{\psi^2} - \frac{|OB_p - OB_q|}{\phi^2} - \frac{|OP_p - OP_q|}{\phi^2}) \tag{12}$$

where Γ represents a truncated threshold that is set to 10; ψ and ϕ is set to 1/9 and 1 respectively; OB is an occlusion boundary mask from the occlusion boundary map OBM and OP is an occluded pixel mask from the occluded pixel map OPM that are enforced as constraints. If an occlusion boundary exists in-between two pixels or one of two neighbouring pixels is an occluded pixel, the strength of smoothness will be reduced. Besides, the color in the view R_{s^*,t^*}, is encoded as a constraint in which two pixels with different colors will decrease smoothness. To solve the proposed occlusion-aware energy function, the graph cut algorithm [Kolmogorov and Zabih, 2002] is used. As a consequence, the proposed occlusion metrics in the energy function especially help a lot to avoid over-smoothing, hence preserving sharp edges, see Fig. 5.

4 EXPERIMENTAL RESULTS

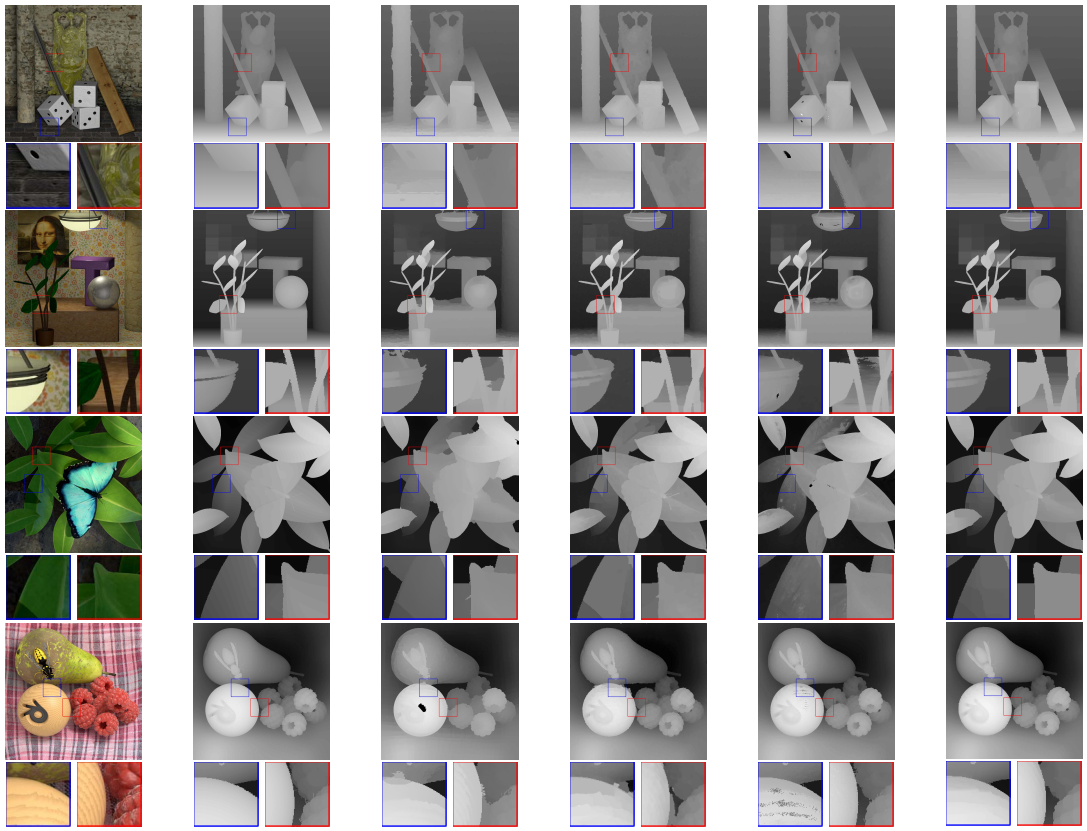
We present the results of the proposed approach that are evaluated on light field datasets, which are composed of synthetic datasets and natural datasets. In order to validate the accuracy and scalability, the experimental results are compared with several state-of-the-art references, PSD [Jeon et al., 2015], OADE [Wang et al., 2015], SPO [Zhang et al., 2016], and an Enhanced Depth Estimation Reference Software eDERS [Senoh et al., 2018]. Note that the results from the state-of-the-art references are generated by utilizing their public code under default settings, except for the number of labels and the disparity range. For validations onto both datasets, two metrics are adopted: a direct metric RMSE for synthetic datasets with available ground truth, and an indirect metric for natural datasets without ground truth (i.e, view synthesis quality using the estimated depth maps).

4.1 Synthetic Dataset

A popular synthetic dataset HCI [Wanner et al., 2013] with ground truth is used for qualitative and quantitative comparisons. Note that the number of labels and the disparity range used into the state-of-the-art works are set to the same values with the proposed method for the sake of better comparisons. The HCI dataset includes 9x9 densely-sampled light field views with a low resolution and has quite a low disparity range, i.e., less than 8 pixels. To well estimate the disparities, 101 disparity labels (a label is less than 8/100 pixel) are employed in all four approaches. Table 1 illustrates that we achieve the highest accuracy of disparity maps on this dataset when compared with PSD, SPO and OADE. Herein, the RMSE for the central view of light fields is calculated as done in [Chen et al., 2014, Wang et al., 2015, Zhu et al., 2017] thanks to the given ground truth. Fig. 6 shows our visual comparisons against the ground truth and the three references. From this comparison, we clearly observe that our framework produces the closest disparity maps to the ground truth with good disparity discontinuity preservations.

Dataset	Buddha	Buddha2	Horses	Medieval	MonasRoom	Papillon	StillLife	Average
PSD	0.109	**0.071**	0.151	0.125	0.084	0.248	0.294	0.154
OADE	0.098	0.109	0.146	0.115	0.088	0.108	0.199	0.123
SPO	0.076	0.101	0.113	**0.094**	0.075	**0.081**	0.119	0.094
Proposed	**0.057**	**0.071**	**0.072**	0.099	**0.072**	0.088	**0.103**	**0.080**

Table 1: The Root Mean Squared Error (RMSE), the lowest value in bold black means the highest accuracy.

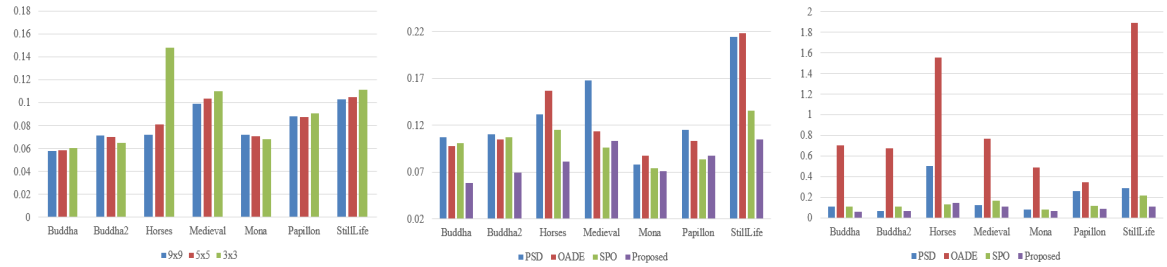


(a) Central View (b) Ground truth (c) PSD (d) OADE (e) SPO (f) Proposed

Figure 6: Disparity estimation results on the HCI dataset. From the top to the bottom, it corresponds to the scene 'Buddha', 'MonasRoom', 'Papillon', 'StillLife' respectively. Our disparity maps seem less noisy than SPO and less over-smoothed than PSD and OADE at occlusion boundary regions, see close-ups of 'StillLife', 'Buddha', etc.

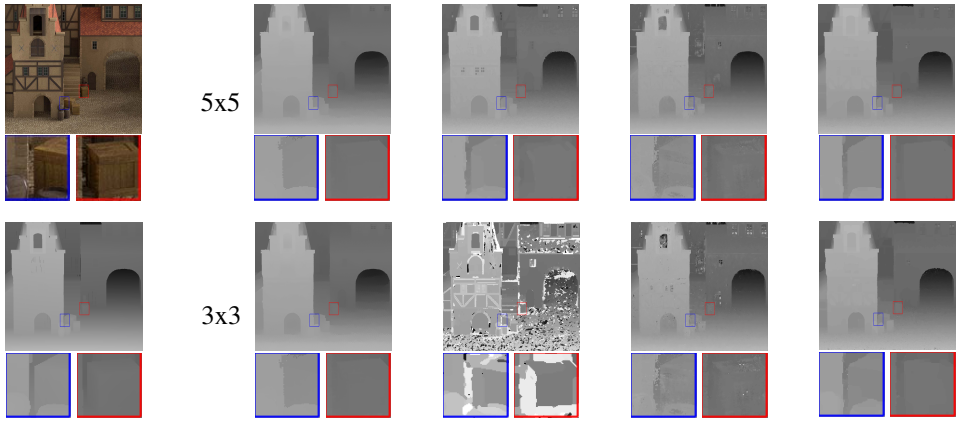
4.2 Density and Disparity Range

When the light field is sparsely-sampled with a large disparity range, it might pose a challenge for the state-of-the-art methods. Hence, we explore the performance of the proposed method on such light fields, which are obtained by skipping a multiple of 2 views from the 9x9 views in both angular directions (i.e., the 5x5 and 3x3 light fields in the paper). Similar to Sec 4.1, the RMSE is calculated for the 5x5 and 3x3 light fields. The computed RMSE is firstly made comparisons with that in Sec 4.1. We can see from Fig. 7 (a) that the errors seem almost unchanged except in the scene 'Horses'. Furthermore, we compare the proposed results with the state-of-the-art references, which is shown in Fig. 7 (b) and (c). It demonstrates that the proposed method, in contrast, mostly achieves the lowest errors and exhibits the robustness to the density and disparity range of the light field. Our method, meanwhile, get at least 14.9%, 19.6%, 29.3% RMSE gains on average in the 9x9, 5x5 and 3x3 light fields respectively. Fig. 8, Fig. 9 and Fig. 10 illustrate the visual comparision results on the 'Medieval', 'Papillon' and 'StillLife' scene respectively. We observe that the quality of the disparity map from OADE [Wang et al., 2015] degrades gradually with a smaller number of light field views, whereas the PSD [Jeon et al., 2015] and SPO [Zhang et al., 2016] decrease a bit but more than that of the proposed method. Moreover, the proposed method does not behave more smoothed as PSD [Jeon et al., 2015] or more noised as SPO [Zhang et al., 2016]. Therefore



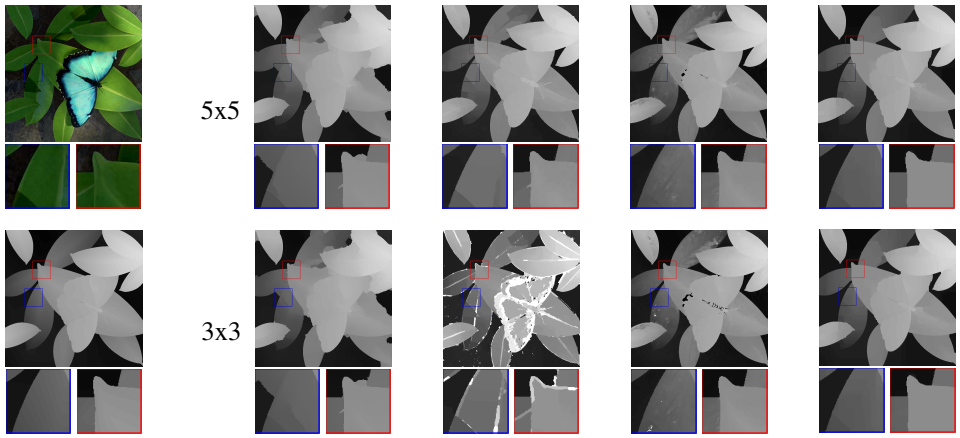
(a) Proposed (b) 5x5 (c) 3x3

Figure 7: The RMSE of the proposed framework in the 9x9, 5x5 and 3x3 light fields is shown in (a). (b) and (c) show the RMSE comparisons between the proposed and the state-of-the-art references in the 5x5 and 3x3 light fields respectively. The lowest value means the highest accuracy.



(a) Central view and Ground truth (b) PSD (c) OADE (d) SPO (e) Proposed

Figure 8: Disparity estimation results on 'Medieval'. Our disparity map is robust around the edges of the wall and/or the box.



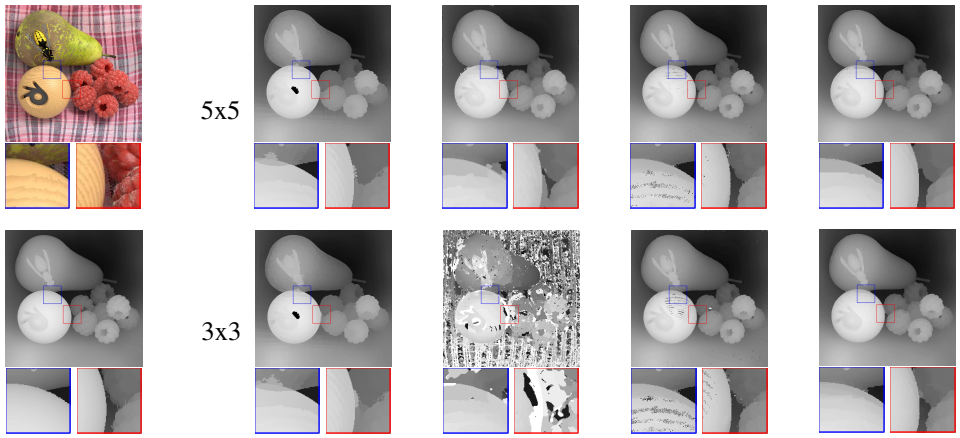
(a) Central view and Ground truth (b) PSD (c) OADE (d) SPO (e) Proposed

Figure 9: Disparity estimation results on 'Papillon'. Our disparity map is achieved with a preciser disparity discontinuity and without noise, see the edges of the leaves.

our method is scalable to the density and disparity range of the light field.

4.3 Occlusion Boundary

Since the accurate occlusion detections were integrated into our global optimization, the occlusion boundary map *OBM* extracted from the final disparity map *FDM* has a significantly high precision. Table 2 gives our performance against the-state-of-the-art methods on the HCI dataset (9x9 light fields) using the common metric Precison-Recall [Sundberg et al., 2011]. An edge detector is used for extracting the proposed and the ground truth occlusion boundary. From the quantitative value, we learn that the precisest occlusion boundaries on av-



(a) Central view and Ground truth (b) PSD (c) OADE (d) SPO (e) Proposed

Figure 10: Disparity estimation results on 'StillLife'. Our disparity map is robust around the surface of the ball.

Dataset	Buddha	Buddha2	Horses	Medieval	MonasRoom	Papillon	StillLife	Average
OADE	0.6632	0.7515	0.7617	0.6043	0.7469	0.7965	0.6181	0.7086
PSD	0.5536	0.7355	0.7354	0.5719	0.6831	0.6089	0.5145	0.6290
SPO	0.6927	0.8330	0.7642	**0.6894**	0.7449	**0.8352**	**0.7115**	0.7530
Proposed	**0.7719**	**0.8480**	**0.8409**	0.6240	**0.7786**	0.7818	0.6950	**0.7629**

Table 2: The Precision-measure of occlusion boundaries, the highest value means the highest accuracy.

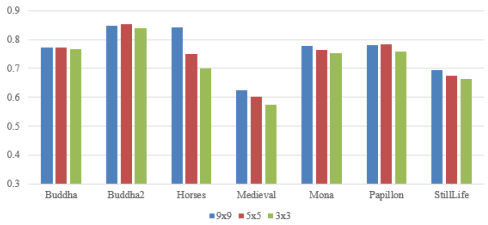


Figure 11: The precisions of the proposed occlusion boundary results onto the 9x9, 5x5 and 3x3 light fields respectively.

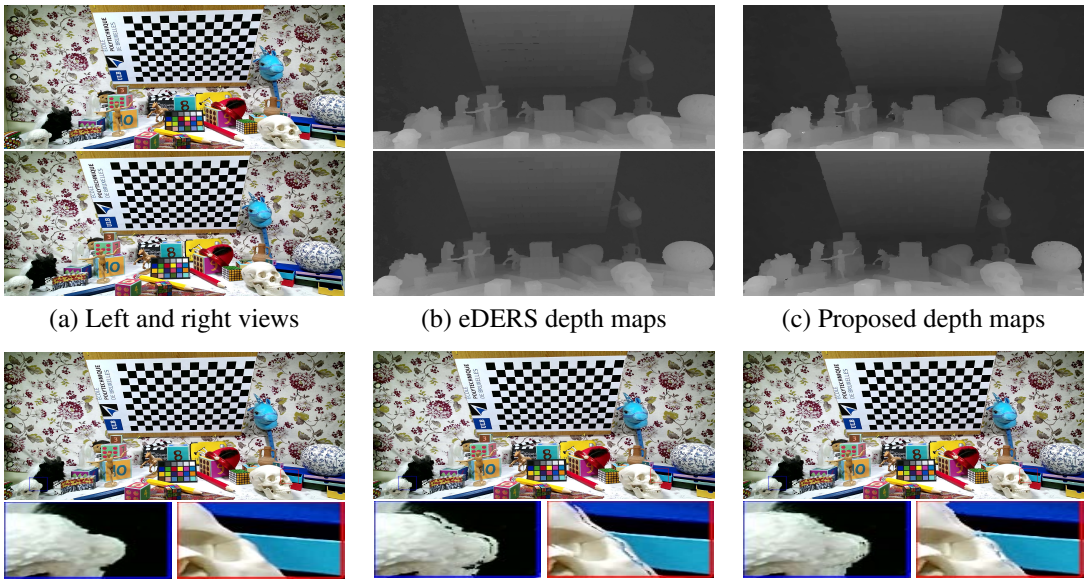
erage are obtained by the proposed work. In addition, the precison values for 5x5 and 3x3 views are also calculated. Note that the 5x5 and 3x3 light fields are also obtained by skipping a multiply of 2 views in both angular directions, similar to Sec 4.2. In Fig. 11, when the number of light field views is reduced, the precision of the occlusion boundary decreases by a very small value, illustrating that the proposed method is also scalable to occlusions in multiple densities and disparity ranges of light fields.

4.4 Natural dataset

In addition to synthetic datasets, the challenging natural datasets ULB Unicorn [Bonatto et al., 2017] and Technicolor Painter [Sabater et al., 2017], which have a larger baseline (35 and 70 mm resp.) for objects at a distance of 0.5 to 4m and a fewer number of views (5x5 and 4x4 views resp.), are evaluated. Moreover, in these datasets, there exists a larger disparity range, i.e., [16-76] and [30, 90] in pixels respectively. Since these two datasets lack ground truth disparities, the view synthesis results generated from view synthesis reference software [Stankiewicz et al., 2013] are used for evaluations, apart from visual comparisons on depth maps that are simply converted from disparity maps. For the view synthesis, two depth maps from two views are required. As the OADE, PSD and SPO are used to predict the disparities for the central view of light fields, we compare our technique with another state-of-the-art technique eDERS [Senoh et al., 2018] (using the same number of disparity labels 241). The experimental results show that the better synthesized/virtual maps are produced from our technique, especially at occlusion regions. Fig. 12 shows that the synthesized map using the proposed depth maps looks much cleaner, see the close-ups in (e) and (f). In Fig. 13, (b) and (c) clearly show that our method correctly estimates the wooden stand and the chair disparities whereas this fails in eDERS. Furthermore, the synthesized map gets more benefits from the proposed depth maps than from the eDERS depth maps, see the close-ups in (e) and (f).

5 CONCLUSIONS

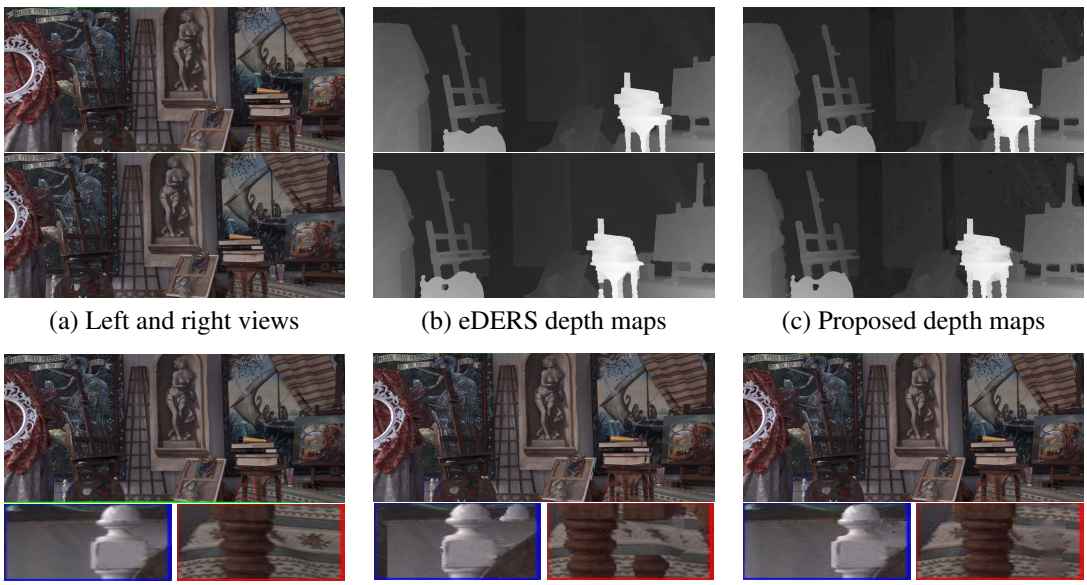
An occlusion-aware framework via multiple edge cues and score updates is proposed for disparity estimation in light fields. Through a variety of evaluations, the proposed method achieves a higher accuracy of disparity estimation on both synthetic and natural datasets when compared with the state-of-the-art approaches. Moreover, the fidelity of the disparity map is still kept even in a sparse light field with a large disparity range.



(a) Left and right views (b) eDERS depth maps (c) Proposed depth maps

(d) Reference/Central view (e) eDERS synthesized map (f) Proposed synthesized map

Figure 12: Depth map and view synthesis result comparisons on the ULB Unicorn dataset. From the top to bottom in (a-c), they correspond to the left and right camera view respectively.



(a) Left and right views (b) eDERS depth maps (c) Proposed depth maps

(d) Reference/Central view (e) eDERS synthesized map (f) Proposed synthesized map

Figure 13: Depth map and view synthesis result comparisons on the Technicolor Painter dataset. From the top to bottom in (a-c), they correspond to the left and right camera view respectively.

6 ACKNOWLEDGMENTS

This work is supported by the China Scholarship Council (CSC) and by Innoviris, the Brussels Institute for Research and Innovation Belgium, under contract number 2015-DS-39a, 3DLicorneA.

7 REFERENCES

[Bonatto et al., 2017] Bonatto, D., Schenkel, A., Lenertz, T., Li, Y., and Lafruit, G. (2017). ULB High Density 2D/3D Camera Array data set, version 2.

[Canny, 1986] Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):679–698.

[Chen et al., 2014] Chen, C., Lin, H., Yu, Z., Kang, S. B., and Yu, J. (2014). Light field stereo matching using bilateral statistics of surface cameras. In *Computer Vision and Pattern Recognition*

(CVPR), 2014 IEEE Conference on, pages 1518–1525. IEEE.

[He et al., 2010] He, K., Sun, J., and Tang, X. (2010). Guided image filtering. In *European conference on computer vision*, pages 1–14. Springer.

[Huang et al., 2017] Huang, J., Chen, Z., Ceylan, D., and Jin, H. (2017). 6-DOF VR videos with a single 360-camera. In *Virtual Reality (VR), 2017 IEEE*, pages 37–44. IEEE.

[Jeon et al., 2015] Jeon, H.-G., Park, J., Choe, G., Park, J., Bok, Y., Tai, Y.-W., and So Kweon, I. (2015). Accurate depth map estimation from a lenslet light field camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1547–1555.

[Kim et al., 2013] Kim, C., Zimmer, H., Pritch, Y., Sorkine-Hornung, A., and Gross, M. H. (2013). Scene reconstruction from high spatio-angular resolution light fields. *ACM Trans. Graph.*, 32(4):73–1.

[Kolmogorov and Zabih, 2002] Kolmogorov, V. and Zabih, R. (2002). Multi-camera scene reconstruction via graph cuts. In *European conference on computer vision*, pages 82–96. Springer.

[MPEG-I, 2017] MPEG-I (2017). Coded representation of immersive media. *MPEG-I Part 3: immersive video, ISO/IEC JTC1/SC29 NP 23090-3*.

[Ng et al., 2005] Ng, R., Levoy, M., Brédif, M., Duval, G., Horowitz, M., and Hanrahan, P. (2005). Light field photography with a hand-held plenoptic camera. *Computer Science Technical Report CSTR*, 2(11):1–11.

[Sabater et al., 2017] Sabater, N., Boisson, G., Vandame, B., Kerbiriou, P., Babon, F., Hog, M., Gendrot, R., Langlois, T., Bureller, O., Schubert, A., et al. (2017). Dataset and pipeline for multi-view lightfield video. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1743–1753. IEEE.

[Scharstein and Szeliski, 2002] Scharstein, D. and Szeliski, R., "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International journal of computer vision*, vol. 47, no. 1-3, pp. 7–42, 2002.

[Senoh et al., 2018] Senoh, T., Yamamoto, K., Tetsutani, N., and Yasuda, H. (2018). Enhanced DERS for Quad Reference Views (eDERS). *ISO/IEC JTC1/SC29/WG11 MPEG2018 m41955*.

[Stankiewicz et al., 2013] Stankiewicz, O., Wegner, K., Tanimoto, M., and Domanski, M. (2013). Enhanced view synthesis reference software (VSRS) for free-viewpoint television. *ISO/IEC JTC*, 1:533–541.

[Stutz et al., 2018] Stutz, D., Hermans, A., and Leibe, B. (2018). Superpixels: an evaluation of the state-of-the-art. *Computer Vision and Image Understanding*, 166:1–27.

[Sundberg et al., 2011] Sundberg, P., Brox, T., Maire, M., Arbeláez, P., and Malik, J. (2011). Occlusion boundary detection and figure/ground assignment from optical flow. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2233–2240. IEEE.

[Wang et al., 2015] Wang, T.-C., Efros, A. A., and Ramamoorthi, R. (2015). Occlusion-aware depth estimation using light-field cameras. In *Computer Vision (ICCV), 2015 IEEE International Conference on*, pages 3487–3495. IEEE.

[Wanner and Goldluecke, 2012] Wanner, S. and Goldluecke, B. (2012). Globally consistent depth labeling of 4D light fields. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 41–48. IEEE.

[Wanner and Goldluecke, 2014] Wanner, S. and Goldluecke, B. (2014). Variational light field analysis for disparity estimation and super-resolution. *IEEE transactions on pattern analysis and machine intelligence*, 36(3):606–619.

[Wanner et al., 2013] Wanner, S., Meister, S., and Goldluecke, B. (2013). Datasets and benchmarks for densely sampled 4d light fields. In *VMV*, pages 225–226. Citeseer.

[Wei and Quan, 2005] Wei, Y. and Quan, L. (2005). Asymmetrical occlusion handling using graph cut for multi-view stereo. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 902–909. IEEE.

[Zhang et al., 2010] Zhang, C., Wang, L., and Yang, R. (2010). Semantic segmentation of urban scenes using dense depth maps. In *European Conference on Computer Vision*, pages 708–721. Springer.

[Zhang et al., 2016] Zhang, S., Sheng, H., Li, C., Zhang, J., and Xiong, Z. (2016). Robust depth estimation for light field via spinning parallelogram operator. *Computer Vision and Image Understanding*, 145:148–159.

[Zhu et al., 2017] Zhu, H., Wang, Q., and Yu, J. (2017). Occlusion-model guided antiocclusion depth estimation in light field. *IEEE Journal of Selected Topics in Signal Processing*, 11(7):965–978.

ISSN 1213-6972 Journal of WSCG http://www.wscg.eu/ Vol.26, 2018, No.2

A Point-Based and Image-Based Multi-Pass Rendering Technique for Visualizing Massive 3D Point Clouds in VR Environments

Sören Discher
Hasso Plattner Institute
University of Potsdam
Prof.-Dr.-Helmert-Straße 2-3
14482 Potsdam, Germany
soeren.discher@hpi.de

Leon Masopust
Hasso Plattner Institute
University of Potsdam
Prof.-Dr.-Helmert-Straße 2-3
14482 Potsdam, Germany
leon.masopust@student.hpi.de

Sebastian Schulz
Hasso Plattner Institute
University of Potsdam
Prof.-Dr.-Helmert-Straße 2-3
14482 Potsdam, Germany
sebastian.schulz@student.hpi.de

Rico Richter
Hasso Plattner Institute
University of Potsdam
Prof.-Dr.-Helmert-Straße 2-3
14482 Potsdam, Germany
rico.richter@hpi.de

Jürgen Döllner
Hasso Plattner Institute
University of Potsdam
Prof.-Dr.-Helmert-Straße 2-3
14482 Potsdam, Germany
juergen.doellner@hpi.de

ABSTRACT

Real-time rendering for 3D point clouds allows for interactively exploring and inspecting real-world assets, sites, or regions on a broad range of devices but has to cope with their vastly different computing capabilities. Virtual reality (VR) applications rely on high frame rates (i.e., around 90 fps as opposed to 30 - 60 fps) and show high sensitivity to any kind of visual artifacts, which are typical for 3D point cloud depictions (e.g., holey surfaces or visual clutter due to inappropriate point sizes). We present a novel rendering system that allows for an immersive, nausea-free exploration of arbitrary large 3D point clouds on state-of-the-art VR devices such as HTC Vive and Oculus Rift. Our approach applies several point-based and image-based rendering techniques that are combined using a multi-pass rendering pipeline. The approach does not require to derive generalized, mesh-based representations in a pre-processing step and preserves precision and density of the raw 3D point cloud data. The presented techniques have been implemented and evaluated with massive real-world data sets from aerial, mobile, and terrestrial acquisition campaigns containing up to 2.6 billion points to show the practicability and scalability of our approach.

Keywords

Virtual reality, 3D point clouds, Real-time rendering

1 INTRODUCTION

Virtual reality (VR) devices, for example Oculus Rift[1] or HTC Vive[2], open up new ways to present digital 3D models on standard consumer hardware, granting users the perception of being physically present in a 3D virtual environment [1, 2]. In general, the corresponding 3D models can be designed and modeled for a particular purpose (e.g., game environment) or can be derived by captured data from real-world sites or assets (e.g., building models).

[1] https://www.oculus.com/rift/
[2] https://www.vive.com

For complex sites, e.g., buildings with a highly detailed interior, or large areas, e.g., cities and landscapes, manually modeling 3D contents is neither time efficient nor cost efficient due to the required effort [26]. As a remedy, there is a strong demand for methods and techniques that (1) automatically and efficiently capture real-world sites of arbitrary size and complexity with high precision and that (2) directly integrate the resulting 3D contents into VR applications without having to sacrifice any captured details.

In recent years, automatically capturing real-world sites by means of 3D point clouds has become increasingly cost efficient and time efficient due to technological advances in remote and in-situ sensing technology [11]. As an example, active and passive sensing technology,

https://doi.org/10.24132/JWSCG.2018.26.2.2

a *Airborne scan of a city.* b *Terrestrial indoor scan.*

Figure 1: Examples of massive 3D point clouds being immersively visualized using our rendering system and an HTC Vive. Supported interaction techniques include measuring of distances as well as rotating and scaling of the rendered data.

such as *light detection and ranging* (LiDAR), radar, or aerial and digital cameras, provides precision rates of up to a few centimeters or millimeters [17, 27]. By attaching those sensors to moving vehicles, such as cars or unmanned aircraft systems (UAS), large areas can be covered within few hours, resulting in massive data sets containing hundreds of gigabytes of raw data [21, 31].

Large unstructured collections of 3D points, called *3D point clouds*, can be directly used as interactively explorable models by combining *level-of-detail (LoD)* concepts, out-of-core strategies, and external memory algorithms [32, 14]: State-of-the-art rendering systems are capable of handling enormous amounts of 3D point cloud data, e.g., billions of points for a non-immersive inspection and visualization on a multitude of devices with vastly different CPU and GPU capabilities [24, 7]. However, they typically focus on non-immersive applications, carefully balancing the trade-off between rendering quality and performance [36]. In VR applications additional challenges are raised:

- **Stereo rendering**. To generate a stereoscopic image, each scene has to be rendered for two displays simultaneously.
- **High rendering quality**. Visual artifacts such as visible holes between neighboring points or visual clutter tend to be more noticeable on VR displays, can easily break the immersion [37] and, therefore, need to be fixed.
- **High frame rates of 90 fps**. Nausea, i.e., the feeling of motion sickness, typically occurs when the motion-to-photon-latency, i.e., the time required for the depicted images to update after a physical movement by the user, becomes too high. As a remedy, the built-in displays of VR devices such as Oculus Rift or HTC Vive operate at 90 Hz [39]. Hence, frames have to be rendered at a considerably higher speed compared to non-immersive applications, for which frame rates between 30 and 60 fps are usually sufficient.

For these reasons, applications for VR devices frequently have to reduce the precision and density of the data, either by thinning the respective point clouds [29] or by converting them into generalized 3D meshes [3].

In this paper, we present a rendering system (Section 3) that allows for an interactive, immersive, and nausea-free visualization of arbitrary large 3D point clouds on state-of-the-art VR devices (Fig. 1). To that end, we combine selected rendering techniques for 3D point clouds that can be roughly sorted into two categories: *Performance optimization techniques* that speed up the rendering pipeline (Section 4) and *image optimization techniques* that improve the overall image quality (Section 5). All techniques have been implemented and are evaluated for two massive, real-world data sets. We end with a conclusion and an overview of future work.

2 RELATED WORK

3D point clouds are widely used in a variety of geospatial [9] and non-geospatial applications [38], used in diverse areas such as building information modeling [28], urban planning and development [25] or the digital preservation of cultural heritage [24]. As fundamental geospatial data representation, 3D point clouds provide highly detailed geometry information about a site that can be further extracted and leveraged by applying point-based analysis algorithms [6, 8]. Although this not being the focus, our approach provides efficient means to visualize results of those analyses, e.g., by applying different per-point color schemes (Fig. 3).

Presentation and interactive visualization of 3D point clouds have to cope with the corresponding massive amount of data, which generally exceeds available CPU and GPU capabilities [15]. To render massive data sets with billions of points, out-of-core rendering concepts and spatial data structures are required to decouple rendering efforts from data management such as

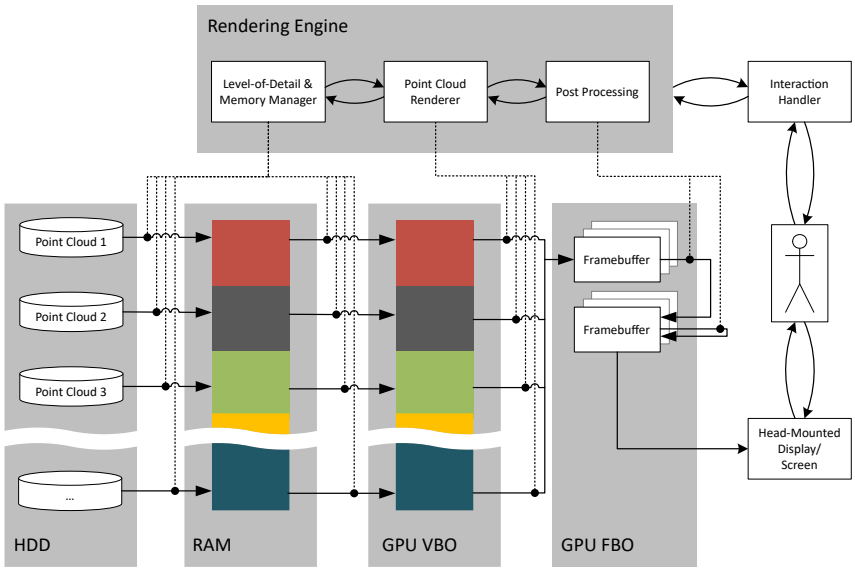


Figure 2: Overview of the rendering pipeline and data flow between hard disk drive (HDD), random-access memory (RAM), vertex buffer objects (VBO), and frame buffer objects (FBO). Different 3D point clouds are rendered separately but share a single memory budget.

quadtrees [13], octrees [12], or kd-trees [14] to subdivide 3D point clouds into small, representative subsets that are suitable for real-time rendering. Out-of-core approaches and web-based rendering concepts are frequently combined. For example, a central server infrastructure can be used to organize and distribute the corresponding 3D point clouds, which limits workload and data traffic on client side [36, 7]. While those approaches allow to visualize massive data sets on client devices with vastly different hardware and graphics capabilities, they generally provide neither visual quality nor rendering performance as required by an immersive visualization.

Real-time rendering is based on performance optimization techniques: While techniques such as view frustum culling and detail culling can be easily applied to 3D point clouds, occlusion, backface, and portal culling are designed with mesh-based geometry and closed surfaces in mind [1]. Due to the unstructured nature of 3D point clouds those techniques require adaptation before being applicable to point-based rendering. Our rendering system implements occlusion culling based on the reverse painter's algorithm [19]. We decided against adapting backface and portal culling as both techniques require specific knowledge or preprocessed information about a 3D point cloud (e.g., per-point normals, semantic information) that might not always be available. Performance optimization techniques specifically for VR applications have been discussed by [39]. Some of those techniques, such as the hidden mesh or the single-pass stereo rendering, are implemented and evaluated by our rendering system.

Visual optimization techniques for 3D point clouds are discussed by several authors, an overview is given by [15]. Visual clutter and holes between neighboring points can be addressed by applying appropriate size, orientation, and color schemes to each point [36, 32]. While leading to good visual results, those techniques also raise the computational cost due to calculating point sizes in object space – either in a pre-processing step [4] or during rendering [30]. As an alternative that scales better for massive data sets, visual artifacts can be eliminated via post-processing using image-based rendering techniques, e.g., to fill holes ([10], [33]), to blur visual clutter [22], or to emphasize depth cues [5, 23]. In the context of VR applications Schütz [37] introduces the usage of point cloud mipmaps as well as multisampling for a reduction of z-fighting and softer edges, which we also evaluate in this paper.

Figure 3: Different color schemes can be applied at runtime. Left: RGB colors extracted from aerial imagery. Right: Colorization based on surface categories.

3 SYSTEM OVERVIEW

Our implementation of the rendering system is based on a multi-pass rendering pipeline (Fig. 2) that can

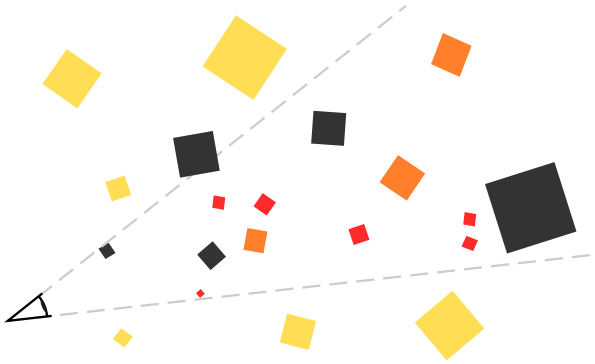


Figure 4: Culling techniques used to reduce the amount of points to be rendered: View frustum culling (yellow), occlusion culling (orange), detail culling (red).

be divided into three distinct stages: Data subset selection, point cloud rendering, and image-based post-processing.

3.1 Level-of-Detail and Data Subset Selection

Instead of rendering every point of a given data set, we determine a *representative subset* of points that can be managed by available CPU and GPU capabilities. Those subsets are determined on a per-frame basis using two major criteria (Fig. 4): First, points outside the current view frustum are excluded as they would not be visible anyway (i.e., *view frustum culling*). Second, points are aggregated based on their spatial position to accommodate for the perspective distortion resulting in areas farther away from the current view position to appear smaller on screen (i.e., *detail culling*). To provide an efficient access to representative data subsets, the 3D point cloud is hierarchically subdivided using a kd-tree, i.e., a binary tree whose splitting planes can be freely positioned alongside the respective coordinate axes. This allows for minimal tree traversal times during rendering as the resulting tree structures are guaranteed to be balanced independently of the data's spatial distribution. For each 3D point cloud a separate kd-tree is generated in a preprocessing step. A flexible memory budget is defined to limit the amount of points that can be rendered per frame. While each 3D point cloud is rendered separately, the memory budget is shared among them. As the performance may vary based on scene complexity and applied rendering techniques, the memory budget is adjusted dynamically to guarantee 90 fps at any time.

3.2 Point Cloud Rendering

Selected data subsets are rendered into so-called g-buffers [34], i.e., specialized frame buffer objects (FBO) that combine multiple 2D textures for, e.g., color, depth, or normal values. This provides efficient means to apply varying post-processing effects that

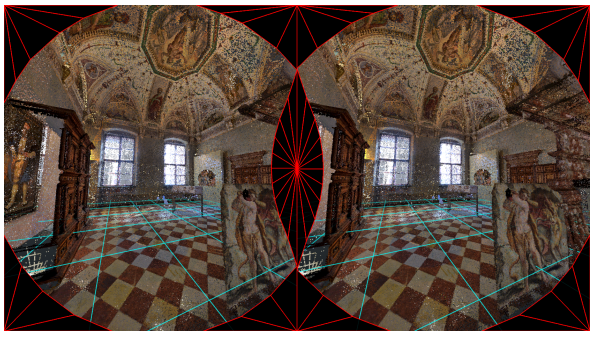


Figure 5: A separately rendered mesh serves as a mask to discard fragments beyond the visible area of an VR device's screens early on.

improve the visual quality of the final image being displayed on the VR device. Furthermore, different rendering techniques for 3D point clouds can be configured, selected, and combined at runtime, allowing to dynamically adjust a 3D point cloud's appearance (e.g., size and color scheme applied to each point) (Fig. 3) as well as the overall rendering performance (Section 4).

3.3 Image-Based Post-Processing

The rendering pipeline's final stage operates recursively on the previously generated g-buffers, allowing to configure and combine several image-based rendering techniques. As an example, rendering techniques for hole-filling, blurring, anti-aliasing as well as edge detecting and highlighting can be efficiently combined to improve the visual quality of the rendering (Section 5).

3.4 Interaction Handling

An interaction handler is responsible for managing user interactions and for updating the visualization accordingly. Users may (1) change view position and angle, (2) configure and select applied rendering techniques and color schemes, (3) measure distances between points, or (4) scale and rotate rendered 3D point clouds.

4 PERFORMANCE OPTIMIZATION TECHNIQUES

To further improve the performance of our rendering system on state-of-the-art VR devices, we have implemented and evaluated three rendering techniques: Hidden mesh rendering, reverse painter's algorithm, and single-pass stereo rendering.

4.1 Hidden Mesh Rendering

Due to the radially symmetric distortion produced by the lenses of an VR device, the actually visible area of the built-in screens is restricted to a circular area (Fig. 5). To prevent unnecessary fragment shader operations, fragments outside that area are discarded early, using a

separately rendered mesh representing the hidden parts of the screen as a mask that is evaluated using early fragment testing [39].

4.2 Reverse Painter's Algorithm

As a GPU-based *occlusion culling* technique (Fig. 4), the reverse painter's algorithm [19] describes efficient means to prevent occluded fragments from being unnecessarily processed by the fragment shader. Based on early fragment testing, scene objects should be rendered in order of their distance to the view position for the technique to have a measurable effect. Calculating such an order on a per-point basis would be inefficient. As each point belongs to a specific node of the kd-tree however, we can instead perform that calculation on a per-node basis, considering only those nodes that have been selected for rendering.

4.3 Single-Pass Stereo Rendering

VR devices require to render all view dependent items from two different views representing the left and right eye, respectively. *Single-pass stereo rendering* aims to reduce the *CPU overhead* by rendering both views in a single render pass [20]. To that end, the frame buffer size is doubled, assigning each half to one eye. Instanced rendering is used to avoid duplicated draw calls. It duplicates each point and applies the corresponding view transform at the vertex shader stage. To minimize the probability of points spilling over into the opposite half of the frame buffer, we apply a heuristic that shrinks points close to the border. Preventing such artifacts completely would require to discard affected fragments explicitly, which would be incompatible to early fragment testing as required by the techniques presented above.

5 IMAGE OPTIMIZATION TECHNIQUES

The immersiveness of a virtual scene is negatively affected by any kind of visual artifacts or inconsistencies one would not expect in the real-world, such as aliasing, z-fighting, and insufficient or missing depth cues [1]. In 3D point cloud depictions, the most noticeable artifacts arise when points representing a continuous surface are sized inappropriately, resulting in either a holey appearance of those surfaces or visual clutter due to overlapping points (Fig. 8a+b). We aim to minimize such artifacts by applying the following rendering techniques: Adaptive point sizes, paraboloid rendering, image-based post-processing (i.e., edge highlighting, blurring, filling), and multisampling.

5.1 Adaptive Point Sizes

The different nodes of LoD data structures exhibit noticeable differences regarding the point density. Thus,

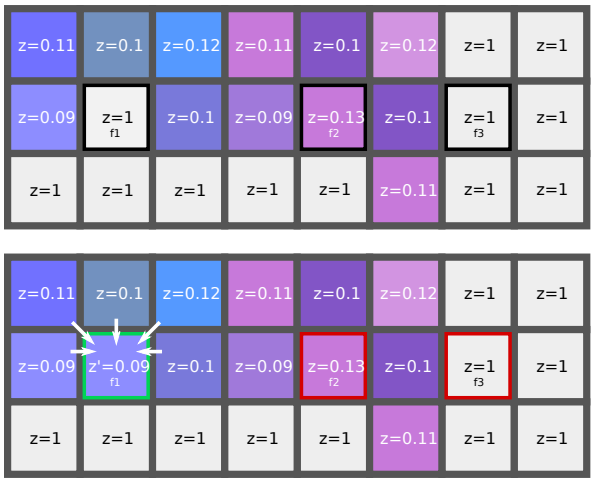


Figure 6: Fragment *f1* is detected as a hole based on depth differences to its neighbors and gets assigned the minimum depth value within its neighborhood; *f2* and *f3* remain unchanged as they fail the distance threshold and the minimum number of neighbors, respectively.

Figure 7: Contrasting color values can be harmonized using blurring to smooth aliasing and z-fighting.

assigning all points a uniform size results in either holes between neighboring points or overlaps and visual clutter. Schütz addresses that issue by adjusting each point's size based on the maximum LoD within its local neighborhood [36]. While we also adjust point sizes adaptively, our technique operates on a per-node instead of a per-point basis, thus, avoiding the need for a separate render pass to calculate each point's LoD. In that regard, our technique is similar to the one proposed by Scheiblauer [35]. However, we use inherently balanced kd-trees in favor of octrees. For each node, we determine its deepest descendant that has been selected for rendering. The adaptive point size for that descendant is then applied to all of its ancestors. Furthermore, we calculate point sizes based on a node's bounding box rather than its LoD since nodes of the same LoD might still feature drastically different point densities. While our technique drastically and effectively reduces holes and overlaps, it does not exclude those artifacts entirely. For example, if nodes selected for rendering form a heavily unbalanced tree, some points might be rendered too small (Fig. 8c). We fill the resulting holes via post-processing.

Figure 8: Incorrectly sized points may lead to a holey appearance (a – point size of 1px) or visual clutter (b – point size of 5px). An adaptive point size strikes a balance between both artifact types, but does not eliminate them completely (c). This can be minimized by applying paraboloid rendering (d – diameter of 5px) or filling (e – 5x5 filter kernel and point size of 1px).

5.2 Paraboloid Rendering

Paraboloid rendering is a technique introduced by Schütz [36] that aims to further reduce visual clutter by rendering points not as flat, screen-aligned disks but as paraboloids oriented towards the view position. By adding a depth offset to fragments based on their distance to the corresponding point's center, undesired occlusions are drastically reduced (Fig. 8d). As this technique requires us to modify depth values at the fragment shader stage however, it is incompatible with early fragment testing and thus for most of the techniques discussed in Section 4.

5.3 Post-Processing

We use several post-processing techniques to further improve the visual quality: Screen space ambient occlusion (SSAO) [23] and eye-dome lighting (EDL) [5] add depth cues and highlight silhouettes, blurring [22] smoothes aliasing and z-fighting (Fig. 7). Furthermore, we fill remaining holes between points representing the same surface (Fig. 8e). To that end, we adapt the tech-

a *Pedestrian view of a mobile mapping scan.* b *Birds-eye view of a mobile mapping scan.* c *Close-up view of a terrestrial indoor scan.*

Figure 9: Scenes used during the performance evaluation.

nique presented by Dobrev et al. [10], applying two one-dimensional filter kernels instead of a single two-dimensional one for a performance speed up. The filter kernel checks a pixel's neighborhood for significant depth differences and overwrites corresponding pixels with interpolated values from those neighbors being closest to the view position (Fig. 6).

5.4 Multisampling

A technique to further smooth aliasing and reduce z-fighting would be multisampling, which provides a smoother color transition between neighboring fragments by sampling them several times. While this technique also reduces the visibility of outliers, we ultimately opted against it as it would require us to render fragments several times, thus, drastically affecting the performance, especially when combined with post-processing effects.

6 PERFORMANCE EVALUATION

We have implemented the presented rendering system using C++, OpenGL, GLSL, and OpenVR[3]. The test system featured an Intel Core i7-5820K CPU, 16 GB main memory (DDR4, 1200 MHz), a GeForce GTX 980 with 4096 MByte device memory(GDDR5, driver version 390.77) as well as an HTC Vive as the output device. Measurements on an Oculus Rift lead to comparable, slightly better results due to the tighter view frustum. The test data sets comprised a mobile mapping scan of an urban area (2.6 billion points) and a terrestrial indoor scan of an individual site (1.5 billion points). The performance evaluation was conducted for three different scenes (Fig. 9): A close up and a zoomed out view of the urban area (Scene 1 and 2) as well as a close up view of the individual site (Scene 3). We disabled the dynamic memory budget, which guarantees the constant framerate of at least 90 fps, for the evaluation to ensure the comparability of the measured values.

Both *hidden mesh* and *reverse painter's* algorithm improve the rendering performance. However, their effectiveness varies, depending on the number of affected fragments (Table 1). *Single-pass stereo rendering* proved to be less effective as the primary rendering bottleneck is the GPU, not the CPU. On the contrary, the technique even slows the rendering pipeline as view frustum culling needs to be combined for both eyes, thus notably increasing the amount of unnecessarily rendered points per side. Regarding image optimization techniques, *paraboloid rendering* and *multisampling* -as expected- significantly reduces the rendering performance (Table 2) and thus should only be used, if the z-fighting becomes too prominent and significantly affects the immersion. On the other hand, post-processing effects and adaptive point sizes only have a moderate performance impact. While combining all post-processing techniques would amount to a significant performance drop, doing so will hardly be necessary. As an example, *EDL* and *SSAO* aim for similar effects, whereas *blurring* will only be noticeable in specific scenes, e.g., if color values of neighboring points are inconsistent due to an erroneous capturing process.

7 CONCLUSIONS AND FUTURE WORK

We have presented a point-based and image-based multi-pass rendering technique that allows for visualizing massive 3D point clouds on VR devices in non immersion-breaking quality (i.e., reducing visual artifacts) and at nausea-avoiding frame rates (i.e., around 90 fps). The multi-pass approach offers many degrees of freedom for graphics and application design because the applied rendering techniques can be selected and configured at runtime. We envision

[3] https://github.com/ValveSoftware/openvr

#Rendered points	Scene 1 19.8M	Scene 2 6.9M	Scene 3 11.6M
Default	15.93ms	9.23ms	12.15ms
Hidden Mesh	15.59ms	9.19ms	11.87ms
Reverse Painter's	12.95ms	9.27ms	11.11ms
Single-Pass Stereo	17.48ms	9.82ms	13.54ms

Table 1: Average rendering performance of performance optimization techniques in ms/frame. All test runs include view frustum and detail culling. Dynamic memory budget was disabled to ensure comparability of measured values.

#Rendered points	Scene 1 19.8M	Scene 2 6.9M	Scene 3 11.6M
Default	12.82ms	9.21ms	10.77ms
Adaptive Pt. Size	13.88ms	9.48ms	12.46ms
SSAO		+ 2.67 ms	
EDL		+ 0.32 ms	
Filling		+ 1.07 ms	
Blurring		+ 2.17 ms	
Multisampling	17.91ms	10.14ms	16.94ms
Paraboloids Def.	12.72ms	10.77ms	10.26ms
Paraboloids	15.17ms	18.45ms	15.62ms

Table 2: Average rendering performance of image optimization techniques in ms/frame. For paraboloids, hidden mesh rendering and the reverse painter's algorithm were deactivated and an oversized point size (5 px) was used. Dynamic memory budget was disabled to ensure comparability of measured values.

the presented approach to be highly beneficial for applications in the fields of digital documentation, preservation, and presentation of natural and cultural heritage as it allows users to remotely explore and inspect *digital twins* of endangered or hardly accessible sites in a much more immersive way than existing solutions [24]. In building information modeling or urban planning and development, it facilitates planning processes by providing efficient means to integrate additional, mesh-based geometry such as 3D floor plans or building models into the generated stereoscopic 3D point cloud depictions. Tests on data sets with up to 2.6 billion points show the feasibility and scalability of our rendering system.

Future work could focus on performance improvements by distributing the stereo rendering across two separate GPUs as proposed by [40]. To support hardware that is not specifically designed for VR, we plan to integrate web-based rendering concepts for thin clients [18, 16]. Using a centralized server to generate and distribute stereoscopic images would support VR applications on mobile devices with limited CPU and GPU capabilities. In addition, many applications require more sophisticated interaction techniques such as placing annotations or directly manipulating data subsets. We plan to investigate how such interaction techniques can be integrated into the presented rendering system.

ACKNOWLEDGEMENTS

We thank Felix Thiel for his contributions to the implementation. Data has been provided by *virtualcitySYSTEMS*, *SHH Sp. z.o.o*, and *illustrated architecture*.

8 REFERENCES

[1] Akenine-Möller, T., Haines, E., Hoffman, N. Real-time rendering (4th ed.). CRC Press, 2018.

[2] Berg, L.P., Vance, J.M. Industry use of virtual reality in product design and manufacturing: a survey. Virtual reality 21, No. 1, pp.1–17, 2017.

[3] Berger, M., Tagliasacchi, A., Seversky, L., Alliez, P., Guennebaud, G., Levine, J., Sharf, A., Silva, C. A survey of surface reconstruction from point clouds. Computer Graphics Forum 36, No.1, pp. 301–329, 2017.

[4] Botsch, M., Kobbelt, L. High-quality point-based rendering on modern GPUs. In Proc. Pacific Graphics, pp. 335–343, 2003.

[5] Boucheny, C. Interactive Scientific Visualisation of Large Datasets: Towards a Perception-based Approach. PhD thesis, Université Joseph Fourier, 2009.

[6] Boulch, A., Saux, B.L., Audebert, N. Unstructured point cloud semantic labeling using deep segmentation networks. In Proc. 3DOR, pp. 17–24, 2017.

[7] Butler, H., Finnegan, D.C., Gadomski, P.J., Verma, U.K. Plas.io: Open Source, Browser-based WebGL Point Cloud Visualization. In AGU Fall Meeting Abstracts, 2014.

[8] Chen, D., Wang, R., Peethambaran, J. Topologically aware building rooftop reconstruction from airborne laser scanning point clouds. IEEE TGRS 55, No. 12, pp. 7032-7052, 2017.

[9] Cura, R., Perret, J., Paparoditis, N. A scalable and multi-purpose point cloud server (PCS) for easier and faster point cloud data management and processing. ISPRS P& RS 127, pp. 39–56, 2017.

[10] Dobrev, P., Rosenthal, P., Linsen, L. An image-space approach to interactive point cloud rendering including shadows and transparency. Computer Graphics and Geometry 12, No.3, pp. 2–25, 2010.

[11] Eitel, J.U., Höfle, B., Vierling, L.A., Abellán, A., Asner, G.P., Deems, J.S., Glennie, C.L., Joerg, P.C., LeWinter, A.L., Magney, T.S., Mandlburger,

G. Beyond 3-D: The new spectrum of lidar applications for earth and ecological sciences. Remote Sensing of Environment 186, pp. 372–392, 2016.

[12] Elseberg, J., Borrmann, D., Nüchter, A. One billion points in the cloud–an octree for efficient processing of 3D laser scans. ISPRS P & RS 76, pp. 76–88, 2013.

[13] Gao, Z., Nocera, L., Wang, M., Neumann, U. Visualizing aerial LiDAR cities with hierarchical hybrid point-polygon structures. In Proc. GI, pp. 137–144, 2014.

[14] Goswami, P., Erol, F., Mukhi, R., Pajarola, R., Gobbetti, E. An efficient multi-resolution framework for high quality interactive rendering of massive point clouds using multi-way kd-trees. The Visual Computer 29, No. 1, pp. 69–83, 2013.

[15] Gross, M., Pfister, H. (Eds.). Point-based graphics. Morgan Kaufmann, 2011.

[16] Gutbell, R., Pandikow, L., Coors, V., Kammeyer, Y. A framework for server side rendering using OGC's 3D portrayal service. In Proc. Web3D, pp. 137–146, 2016.

[17] Hämmerle, M., Höfle, B., Fuchs, J., Schröder-Ritzrau, A., Vollweiler, N., Frank, N. Comparison of kinect and terrestrial lidar capturing natural karst cave 3-d objects. IEEE GRSL 11, No. 11, pp. 1896–1900, 2014.

[18] Hagedorn, B., Thum, S., Reitz, T., Coors, V., Gutbell, R. OGC 3D Portrayal Service 1.0, OGC Implementation Standard 1.0, Open Geospatial Consortium, 2017.

[19] Hughes, J.F., van Dam, A., McGuire, M., Sklar, D.F., Foley, J.D., Feiner, S.K., Akeley, K. Computer graphics: principles and practice (3rd ed.), Addison-Wesley Professional, 2013.

[20] Johansson, M. Efficient stereoscopic rendering of building information models (BIM). JCGT 5, No.3, 2016.

[21] Langner, T., Seifert, D., Fischer, B., Goehring, D., Ganjineh, T., Rojas, R. Traffic awareness driver assistance based on stereovision, eye-tracking, and head-up display. In Proc. IEEE ICRA, pp. 3167–3173, 2016.

[22] Lukin, A. Tips & tricks: Fast image filtering algorithms. In Proc. GraphiCon, pp. 186–189, 2016.

[23] Mittring, M. Finding next gen: Cryengine 2. In ACM SIGGRAPH courses, pp. 97–121, 2007.

[24] Martinez-Rubi, O., de Kleijn, M., Verhoeven, S., Drost, N., Attema, J., van Meersbergen, M., van Nieuwpoort, R., de Hond, R., Dias, E., Svetachov, P. Using modular 3D digital earth applications based on point clouds for the study of complex sites. Intl. Journal of Digital Earth 9, No. 12, pp. 1135–1152, 2016.

[25] Musialski, P., Wonka, P., Aliaga, D.G., Wimmer, M., Gool, L.V., Purgathofer, W. A survey of urban reconstruction. Computer Graphics Forum 32, No. 6, pp. 146–177, 2013.

[26] Nebiker, S., Bleisch, S., Christen, M. Rich point clouds in virtual globes–A new paradigm in city modeling?. Computers, Environment and Urban Systems 34, No. 6, pp. 508–517, 2010.

[27] Ostrowski, S., Jozkow, G., Toth, C., Vander Jagt, B. Analysis of point cloud generation from UAS images. ISPRS Annals 2, No. 1, pp. 45–51, 2014.

[28] Pătrăucean, V., Armeni, I., Nahangi, M., Yeung, J., Brilakis, I., Haas, C. State of research in automatic as-built modelling. Advanced Engineering Informatics 29, No. 2, pp. 162–171, 2015.

[29] Peters, R., Ledoux, H. Robust approximation of the Medial Axis Transform of LiDAR point clouds as a tool for visualisation. Computers & Geosciences 90, pp. 123–133, 2016.

[30] Preiner, R., Jeschke, S., Wimmer, M. Auto Splats: Dynamic Point Cloud Visualization on the GPU. In Proc. EGPGV, pp. 139–148, 2012.

[31] Remondino, F., Spera, M.G., Nocerino, E., Menna, F., Nex, F., Gonizzi-Barsanti, S. Dense image matching: comparisons and analyses. In Proc. DigitalHeritage, pp. 47–54, 2013.

[32] Richter, R., Discher, S., Döllner, J. Out-of-core visualization of classified 3d point clouds. 3D Geoinformation Science, pp. 227–242, 2015.

[33] Rosenthal, P., Linsen, L. Image-space point cloud rendering. In Proc. CGI, pp. 137–143, 2008.

[34] Saito, T. and Takahashi, T. Comprehensible Rendering of 3-D Shapes. In Proc. SIGGRAPH Computer Graphics, pp. 197–206, 1990.

[35] Scheiblauer, C., Pregesbauer, M. Consolidated Visualization of Enormous 3D Scan Point Clouds with Scanopy. In Proc. CHNT, pp. 242–247, 2011.

[36] Schütz, M. Potree–Rendering Large Point Clouds in Web Browsers. Master thesis, Technische Universität Wien, 2016.

[37] Schütz, M. Massive Time-Lapse Point Cloud Rendering in Virtual Reality. Presentation at SIGGRAPH, 2016.

[38] Sitek, A., Huesman, R.H., Gullberg, G.T. Tomographic reconstruction using an adaptive tetrahedral mesh defined by a point cloud. IEEE T-MI 25, No. 9, pp. 1172–1179, 2006.

[39] Vlachos, A. Advanced VR Rendering. Presentation at GDC, 2015.

[40] Vlachos, A. Advanced VR Rendering Performance. Presentation at GDC, 2016.

ISSN 1213-6972 Journal of WSCG http://www.wscg.eu/ Vol.26, 2018, No.2

Mesh-based Multi-view Normal Integration with Energy Minimization Using Surface Reflectance Properties

Wichayut Eaksarayut
Department of Computer Engineering,
Faculty of Engineering,
Chulalongkorn University
Bangkok, 10330, Thailand
wichayut.ea@student.chula.ac.th

Borom Tunwattanapong
Ratchathani University
Ubon Ratchathani, 34000, Thailand
borom@rtu.ac.th

Pitchaya Sitthi-amorn
Department of Computer Engineering,
Faculty of Engineering,
Chulalongkorn University
Bangkok, 10330, Thailand
pitchaya@cp.eng.chula.ac.th

Nuttapong Chentanez
Department of Computer Engineering,
Faculty of Engineering,
Chulalongkorn University
Bangkok, 10330, Thailand
nuttapong26@gmail.com

ABSTRACT

We propose a technique to reconstruct a general 3D object using surface reflectance information from multiple viewpoints. Our core optimization framework uses multi-view normal integration, which can recovers water-tight surface of the object iteratively in a coarse to fine manner. The integration requires normal vector field from multiple viewpoints, which we can derive from surface reflectance. We then handle the topological changes if self-intersection occurs from the optimization. We also employ the idea of multi-resolution and weighted data heuristic which helps dealing with noisy data and improves both accuracy and optimization time. Our experiment shows that the framework is able to robustly recover 3D surface well with both synthetic and real data.

Keywords

3D Reconstruction, Multi-view normal integration, Multi-view vision, Triangle mesh-based surface

1 INTRODUCTION

3D reconstruction has been widely focused in the field of computer graphics and visions with various applications in today's life, such as medical, engineering, advertisement, and entertainment. This influences researchers to develop new techniques to solve this problem more efficiently and with higher accuracy. Existing state-of-the-art algorithms can reconstruct 3D objects with great accuracy, however they typically cannot handle surface that consist if both highly diffuse and highly specular parts. We leverage recent acquisition techniques that can accurately capture surface normal vector and specular reflection vector [17], and focus our 3D reconstruction algorithm base on normal integration.

There has been a considerable amount of researches that studied the multi-view normal integration problem [5, 15, 18]. Chang et al. [5] is the first to proposed the energy functional for multi-view normal integration that is derived from the classical single view shape-from-shading problem [13] and variational framework has been used in most researches to solve this error functional. Techniques above used implicit functions to represent the surface which gives an advantage on topology adaptation while performing mesh deformation. However, accurately representing a 3D object using implicit functions typically require large memory consumption and computation time, as it requires three-dimensional voxels to represent all the surface. [19] proposed an optimization framework which used triangular-mesh to represent the surface. However, they convert their mesh to an implicit surface in order to handle topological changes. This causes the edge length of the mesh to be up to the size of voxels in which fine details can be lost from converting to implicit surface.

https://doi.org/10.24132/JWSCG.2018.26.2.3

Our technique aims to use multi-view normal integration to reconstruct an arbitrary 3D object using normal and reflectance map from multiple viewpoints. We implemented multi-resolution optimization scheme in our framework which helps the overall optimization converges faster. We applied gradient descent to the error functional and perform all operations directly on the 3D triangle-based mesh. This enables us to control the resolution of the mesh during optimization. However, using this explicit surface representation has its drawbacks. Topology cannot be trivially change and self-intersection may occurs during optimization. We employ the method from [20] to remove self-intersection and handle topological change.

The rest of this paper is organized as follows: we review the related works on Section 2. We define our problem in Section 3. We then explain our proposed method in Section 4, and Section 5 to 6 will be our results and conclusion respectively.

Our main contributions are

- Mesh base optimization scheme that can handle topological change and self-intersection without conversion to implicit representation.
- Multi-resolution optimization.
- Optimization schedule that interleaves matching cost optimization with normal integration.
- Target normal calculation that takes visibility and multi-view information into account and can handle missing data.

2 RELATED WORK

3D reconstruction has gain a lot of attention in computer graphics and computer visions fields. In this section, we will focus on reviewing 3D reconstruction techniques that takes photometric and normal information as their inputs from multiple viewpoints. We refer the reading to an excellent survey for other 3D reconstruction method by Herbort and Wöhler [12].

Early methods for recovering surface information is shape-from-shading [3, 11, 13, 22]. These conventional methods were designed for reconstructing 2.5D surface from a single view information of texture-less object with known light position. Chang et al. [5] introduced a new technique that can reconstructs 3D surface using normal vector information from multiple viewpoints. They proposed their energy functional based on the single-view variational framework for shape-from-shading problem [13]. Geometric PDE is then derived to minimize their proposed functional and level-set method is used as their optimization framework. Recently, Weinmann et al. [18] employed a similar concept of multi-view normal integration in order to reconstruct the surface of high specular object. They calculated the volumetric normal field from projected illumination patterns and then applied global optimization with octree-based min-cut framework. The benefit of using an implicit surface (i.e. level-set, voxels, and octree) as their surface representation is that it automatically handles the topological changes while deforming the surface to the optimal target solution. However, it suffers from a large amount of memory consumption with more detailed mesh and can suffers from slow convergence rate.

A number of previous works uses other surface representation. Esteban et al. [10] refined a visual hull by finding photometric normal consistencies and then deformed their mesh on vertex space. However, problem like self-intersection was not taken into account in their paper. Similarly, Yoshiyasu and Yamazaki [19] used a hybrid framework between intrinsic and extrinsic surface representation by optimizing their energy terms on triangular mesh and convert the mesh into an implicit surface to handles self-intersections. Though, the detail of target mesh can be washed out when converting to implicit surface. Furthermore, Tunwattanapong et al. [17] presented a technique for recovering the geometry of 3D objects by projecting spherical harmonics basis on the object to acquire its reflectance information and then used message passing algorithm on vertex space to minimize their energy functional.

Our proposed method performs optimization directly on triangle mesh similar to [10, 19]. Our energy functional is related to [5], but adding more terms in visibility function to handle inter-reflections and noisy information better. We then minimize our energy functional using gradient descent scheme applying the method from Delaunoy et al. [9] which presented a framework to optimize a triangular mesh with gradient descent scheme. We handle the topological changes by employing similar algorithm from [20, 21]. Their algorithm could fix a mesh with self-intersection without losing details on the other part of the mesh. In addition to surface normal, we used reflectance information as our inputs. This allows our framework to work when the surface is not texture-less Lambertian. Our works compatible with other research in which they measured specularity [17, 18].

3 PROBLEM STATEMENT

The goal of our framework is to recover a full watertight triangular 3D mesh with reflectance information from multiple viewpoints with known intrinsic and extrinsic camera parameters. Our mesh consists of n vertices and m triangles which we denotes our vertices as a matrix $\mathbf{V} = [\mathbf{v}_1 \cdots \mathbf{v}_n]^T$ where $\mathbf{v}_i \in \mathbb{R}^3, i \in [1,n]$ denotes a point in 3D space, and triangle $\mathbf{F} = [\mathbf{f}_1 \cdots \mathbf{f}_m]$ where $\mathbf{f}_j$, $j \in [1,m]$ is a set consists of three adjacent vertices.

Each vertex $\mathbf{v}_i$ has an outward normal $\mathbf{N}(\mathbf{p}_i)$, similarly, each triangle also has its outward normal $\mathbf{N}^{\mathrm{F}}(\mathbf{f}_j)$. Our framework also requires a set of l calibrated cameras $\mathbf{C} = \{\mathbf{c}_1, \cdots, \mathbf{c}_l\}$ located around the target object. Each camera $\mathbf{c}_k$ where $k \in [1 \cdots l]$ has its own intrinsic and extrinsic parameters which can be described as matrices $\mathbf{K}_k$ and $[\mathbf{R}_k|\mathbf{t}_k]$ respectively, where $\mathbf{t}_k$ is a translation vector in $\mathbb{R}^3$ for camera $\mathbf{c}_k$ and the projection from any point $\mathbf{v} \in \mathbb{R}^3$ to image domain of camera $\mathbf{c}_k$ can be written as $\tilde{\mathbf{v}_k} = \mathbf{K}_k[\mathbf{R}_k|\mathbf{t}_k]\mathbf{v}$,$\tilde{\mathbf{v}_k} \in \mathbb{R}^2$. For the simplicity, we will also define this projection function to be $\tilde{\mathbf{v}_k} = \pi_k(\mathbf{v})$ and a lookup function $\nu_{k,\mathbf{X}}(\tilde{\mathbf{v}_k})$ which will return information of image $\mathbf{X}$ at pixel $\tilde{\mathbf{v}_k}$.

We need some information to describe how incident light reflected the object surface which in this case, we use diffuse and specular property of the surface as it is well known and widely used in many research. These information describe how the light reflect from the object surface to the camera lens which we can then use them to optimize the target surface. Our cameras will capture (or synthetically generate) these reflection information separately in each viewpoint. Our research will use four type of reflection data which are, diffuse intensity, diffuse reflection, specular intensity, and specular reflection. These information can then be derived to surface normal and use them in the optimization process which we will elaborate them on Section 4.1.

4 PROPOSED METHOD

In this section, we explain the core algorithm in order to recover water-tight 3D mesh with reflectance information. We perform optimization directly on triangle mesh as in [5]. Therefore, we require an initial surface approximation which can be acquired from various procedures. In our work, we use shape-from-silhouette [14] to compute a visual hull and use them as an initial surface. We assumed that such information is also given as a part of the input data.

We optimize the energy functional in coarse-to-fine manner by implementing multi-resolution optimization. We schedule more optimization iterations at coarse resolution and gradually decrease the optimization iterations in finer resolution iteration. This helps the overall framework to converge faster.

After we have a visual hull, we then minimize the cost functional based on geometric and photometric normal. The concept is to deform the mesh to match the target geometric normal with observed photometric normal.

The input from cameras typically have some noises. We add **target normal blending term** in order to filter out unwanted noise and make the reconstruction more robust and visually appealing.

We minimized our energy functional using a gradient descent scheme on vertex domain (Section 4.1). This is similar to surface evolution on implicit surface framework, instead we evolve our triangular mesh towards the gradient direction directly. This may leads to unwanted self-intersection artifacts. We perform an adaptive remeshing algorithm [20, 21] on self-intersected surface. Our overall procedures is shown in Algorithm 1.

Algorithm 1 Reconstruction Pipeline

```
(V,F) ← Shape-from-silhouette                ▷ Initial shape
for each resolution iteration do
    if mesh is coarse then
        (V,F) ← matchingcost-optimization(V,F)
    for each optimization iteration do
        Find target normal of each V and F
        Calculate ∇E of each V and F
        repeat
            α ← arg min_α E(deform(V,F,∇E,α))
            (V,F) ← deform(V,F,∇E,α)
            (V,F) ← fix-self-intersections(V,F)
        until E(V,F) is converges
    (V,F) ← resample(V,F)
return (V,F)
```

4.1 Multi-view Reflectance Integration

As in prior research about multi-view normal integration [5, 15, 18], we employ an error functional minimization framework based on the conventional shape-from-shading approach [13]. We minimize the cost functional with variational methods by minimizing the disparity of geometric and observed normal fields on the surface domain. In the other words, we evolve the surface so that its geometric normal field matched the observed normal field. However, a normal vector at a point on the given surface can be ambiguous with noisy data which should be taken into account.

We adjusted the multi-view normal field integration functional proposed by Chang et al. [5] which required an initial approximation of surface to integrate with. In this research, we acquired an initial shape approximation using shape-from-silhouette [14] as it is good enough for our algorithm.

From a given initial approximation surface, we refine them by displacing every vertices such that its geometric normal field of both from vertices and triangles matches the observed one. Thus, we define our cost functional of a given vertices $\mathbf{V}$ and triangles $\mathbf{F}$ as follows:

$$E(\mathbf{V},\mathbf{F}) = \sum_{\mathbf{v}\in\mathbf{V}} \omega(\mathbf{v})[1 - (\mathbf{N}_t(\mathbf{v}) \cdot \mathbf{N}_g(\mathbf{v}))] + \sum_{\mathbf{f}\in\mathbf{F}} \omega^{\mathrm{F}}(\mathbf{f})[1 - (\mathbf{N}_t^{\mathrm{F}}(\mathbf{f}) \cdot \mathbf{N}_g^{\mathrm{F}}(\mathbf{f}))] \quad (1)$$

where, $\mathbf{N}_t(\mathbf{v})$ and $\mathbf{N}_t^{\mathrm{F}}(\mathbf{f})$ are observed target normal at vertex $\mathbf{v}$ and triangle $\mathbf{f}$ respectively, $\mathbf{N}_g(\mathbf{v})$ and $\mathbf{N}_g^{\mathrm{F}}(\mathbf{f})$ are geometric normal at vertex $\mathbf{v}$ and triangle $\mathbf{f}$, $\omega(\mathbf{v})$ and $\omega^{\mathrm{F}}(\mathbf{f})$ are a weighting function of vertex $\mathbf{v}$ and triangle $\mathbf{f}$, based on the surface area.

Our reflectance information can be derived to normal vector so that it is consistent with our proposed cost functional. For diffuse component, we can derive them with the following equation:

$$\tilde{\mathbf{N}}_{k,\mathrm{diff}}(\mathbf{p}) = \mathrm{Normalize}\left(\alpha_r \nu_{k,\mathrm{diff}}(\tilde{\mathbf{p}}_k) - \mathbf{p}\right) \tag{2}$$

where $\nu_{k,\mathrm{diff}}(\tilde{\mathbf{p}}_k)$ is a lookup function for diffuse reflection of k$^{\mathrm{th}}$ camera at pixel $\tilde{\mathbf{p}}_k$, $\tilde{N}_{k,\mathrm{diff}}(\mathbf{p})$ is calculated diffuse normal at point $\mathbf{p}$ from camera k, α_r is a radius constant of a projection sphere where the incident light reflected to, and for specular component:

$$\tilde{\mathbf{N}}_{k,\mathrm{spec}}(\mathbf{p}) = \mathrm{Normalize}\left(\frac{\alpha_r \nu_{k,\mathrm{spec}}(\tilde{\mathbf{p}}_k) - \mathbf{p}}{|\alpha_r \nu_{k,\mathrm{spec}}(\tilde{\mathbf{p}}_k) - \mathbf{p}|} + \mathbf{p} - \bar{\mathbf{C}}_k\right) \tag{3}$$

Similarly, where $\nu_{k,\mathrm{spec}}(\tilde{\mathbf{p}}_k)$ is a lookup function for specular reflection, $\tilde{\mathbf{N}}_{k,\mathrm{spec}}(\mathbf{p})$ is calculated specular normal ,and $\bar{\mathbf{C}}_k$ is position vector of the k$^{\mathrm{th}}$ camera.

4.2 Target Normal Calculation

According to (1), there are both $\mathbf{N}_t(\mathbf{v})$ and $\mathbf{N}_t^{\mathrm{F}}(\mathbf{f})$ terms which we need to obtain by observing normal vectors from the photometric information provided. At a vertex $\mathbf{v}$, we calculate the normal vectors from diffuse and specular component separately and blend them with weighting constants as follows:

$$\mathbf{N}_t(\mathbf{v}) = \mathrm{Normalize}(w_{\mathrm{diff}}\mathbf{N}_{t,\mathrm{diff}}(\mathbf{v}) + w_{\mathrm{spec}}\mathbf{N}_{t,\mathrm{spec}}(\mathbf{v})) \tag{4}$$

where w_{diff} and w_{spec} are the weight for diffuse and specular component which can be calculated as follows:

$$w_{\mathrm{diff}} = \sum_{k\in\mathbf{C}} \alpha_{\theta,k}(\mathbf{v})\psi_k(\mathbf{v})\nu_{k,\mathrm{diffalbedo}}(\tilde{\mathbf{v}}_k) \tag{5}$$

$$w_{\mathrm{spec}} = \sum_{k\in\mathbf{C}} \alpha_{\theta,k}(\mathbf{v})\psi_k(\mathbf{v})\nu_{k,\mathrm{specconf}}(\tilde{\mathbf{v}}_k) \tag{6}$$

$$\alpha_{\theta,k}(\mathbf{v}) = \max(0, (-\hat{l}_k \cdot \mathbf{N}_g(\mathbf{v}))) \tag{7}$$

where $\hat{l}_k$ denotes a camera direction vector, $\mathbf{N}_g(\mathbf{v})$ is geometric normal at vertex $\mathbf{v}$, $\psi_k(\mathbf{v})$ is visibility function which will determine if camera $\mathbf{c}_k$ is visible for vertex $\mathbf{v}$. $\nu_{k,\mathrm{specconf}}(\tilde{\mathbf{v}}_k)$ is a look up function for diffuse albedo at point $\mathbf{v}$, and $\nu_{k,\mathrm{specconf}}(\tilde{\mathbf{v}}_k)$ is specular reflection confidence which depends on the acquisition technique.

For each component, we project this point to a set of visible cameras $\mathbf{C}_{\mathrm{seen}}$ and look up for reflectance information. We then use weighted average function based on camera angle towards the surface to calculate for the target normal as follows:

$$\mathbf{N}_{t,\mathrm{diff}}(\mathbf{v}) = \sum_{k\in\mathbf{C}} \alpha_{\theta,k}(\mathbf{v})\psi_k(\mathbf{v})\tilde{\mathbf{N}}_{k,\mathrm{diff}}(\mathbf{v}) \tag{8}$$

$$\mathbf{N}_{t,\mathrm{spec}}(\mathbf{v}) = \sum_{k\in\mathbf{C}} \alpha_{\theta,k}(\mathbf{v})\psi_k(\mathbf{v})\tilde{\mathbf{N}}_{k,\mathrm{spec}}(\mathbf{v}) \tag{9}$$

Similarly, for the triangle case, we used its centroid as a point of projection and then obtain target normal for the triangle.

The visibility terms $\psi_k(\mathbf{v})$ can be easily calculated using ray tracing algorithm like in previous research [5, 9, 15]. However, determining whether the surface in consideration is visible by just ray-tracing might not be enough as there could be some outliers (noise and inter-reflections) which can leads to inaccurate target normal. Therefore, we need to filter such outliers out first by restricting more conditions to visibility terms as follows:

$$\psi_k(\mathbf{v}) = \kappa_m(\mathbf{v})\kappa_p(\mathbf{v})\kappa_{cg}(\mathbf{v})\kappa_{ct}(\mathbf{v})\kappa_{tg}(\mathbf{v}) \tag{10}$$

$$\kappa_m(\mathbf{v}) = \begin{cases} 1, & \text{if } \mathbf{v} \text{ is visible at camera } k \\ 0, & \text{otherwise} \end{cases} \tag{11a}$$

$$\kappa_p(\mathbf{v}) = \begin{cases} 1, & \text{is } \textbf{not} \text{ self-intersected} \\ & \text{along the reflection vector} \\ 0, & \text{otherwise} \end{cases} \tag{11b}$$

$$\kappa_{cg}(\mathbf{v}) = \begin{cases} 1, & -(\hat{l}_i \cdot \mathbf{N}_g(\mathbf{v})) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{11c}$$

$$\kappa_{ct}(\mathbf{v}) = \begin{cases} 1, & -(\hat{l}_i \cdot \nu_{k,\mathrm{spec}}(\tilde{\mathbf{v}}_k)) > 0.5 \\ 0, & \text{otherwise} \end{cases} \tag{11d}$$

$$\kappa_{tg}(\mathbf{v}) = \begin{cases} 1, & \tilde{\mathbf{N}}_{k,\mathrm{spec}}(\mathbf{v}) \cdot \mathbf{N}_g(\mathbf{v}) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{11e}$$

Like in [5, 9, 15], the first term (11a) can be determine by tracing a ray from camera to vertex, if it is not occluded by any surface then this counts as visible. Although from our observation, there are several sources that lead to incorrect target normal acquisition, such as inter-reflections. (11b) checks whether the gathered information is bad from inter-reflection by tracing a ray from position $\mathbf{v}$ along the reflection vector respected to each viewpoint. If the ray hit the mesh itself, we will discard the information and treated this pixel as invalid. The term (11c) checks boundary cases when the ray-tracer hits back-face surface. This can be occurred when tracing to a point located near the silhouette or thin surfaces. For specular component, (11d) filters out the reflection vectors that have wide angle respected to its camera direction vector as the surface that face off the camera are likely to be noisy. Lastly, the term (11e) filters out the bad photometric reflection which face backward respected to the mesh geometry. This mostly occur in the area with inter-reflection.

4.3 Gradient Descent Optimization Scheme

With observed target normal being calculated on every vertices in $\mathbf{V}$ and triangles in $\mathbf{F}$, we then minimize our energy functional in (1) with gradient descent framework. Similar to [9], but with our proposed energy functional. Basically, we will deform our mesh by translating each vertex $\mathbf{v}_i$ along the calculated deformation direction vector $\mathbf{d}_i$ which can be written as:

$$\mathbf{v}_i' = \mathbf{v}_i + t\mathbf{d}_i \tag{12}$$

where $\mathbf{v}_i'$ denotes a deformed vertex $\mathbf{v}_i$ with scalar weight t for direction $\mathbf{d}_i$. This deformation vector can be computed by finding the gradient of energy functional in (1) and energy decreases when the surface is deformed in the opposite gradient direction. Thus, the deformation equation of the whole mesh can be written as:

$$\mathbf{V}' = \mathbf{V} - \beta \nabla E(\mathbf{V}, \mathbf{F}) \tag{13}$$

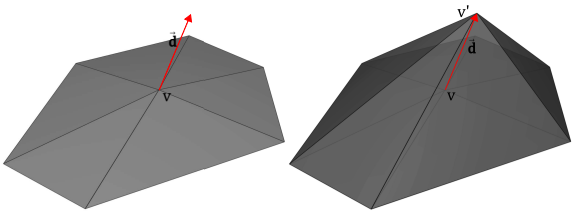


Figure 1: Vertex deformation of point $\mathbf{v}$ toward the direction vector $\mathbf{d}$ which can be calculated by finding gradient of vertex $\mathbf{v}$

Finding the gradient for each vertex $\mathbf{v}_i$ is not trivial, since our energy functional (1) is based on normal terms. Besides, we need to calculate the gradient respect to its position:

$$\nabla E(\mathbf{V}, \mathbf{F}) = \left[\frac{\delta E}{\delta x}(\mathbf{V}, \mathbf{F}), \frac{\delta E}{\delta y}(\mathbf{V}, \mathbf{F}), \frac{\delta E}{\delta z}(\mathbf{V}, \mathbf{F}) \right] \tag{14}$$

Our normal can be derived from its adjacent vertices using the following equations:

$$t_1 = \sum_{i=0}^{k-1} \cos\left(\frac{2\pi i}{k}\right) \mathrm{Adj}(\mathbf{v}, i) \tag{15a}$$

$$t_2 = \sum_{i=0}^{k-1} \sin\left(\frac{2\pi i}{k}\right) \mathrm{Adj}(\mathbf{v}, i) \tag{15b}$$

where $\mathbf{v}$ has k adjacent vertices, t_1 and t_2 are tangent vectors, and $\mathrm{Adj}(\mathbf{v}, i)$ returns the position of $\mathrm{i_{th}}$ adjacent vertex of $\mathbf{v}$. The cross product $t_1 \times t_2$ is then calculated for vertex normal. (For more in details please refer to [16]) With this we can solve for an analytic gradient of the energy with a symbolic differentiation package such as **sympy** [1].

We then perform line search algorithm to find the value β in (13) which will minimize our energy toward the current surface. Then from (13), we have:

$$\arg\min_{\beta} E(\mathbf{V} - \beta \nabla E(\mathbf{V}, \mathbf{F}), \mathbf{F}) = 0 \tag{16}$$

4.4 Target Normal Blending

Some part of the surface may not be captured with high quality information (e.g. highly concave surface) or that part of the surface is totally occluded. This could be problematic as observed target normal vector $\mathbf{N}_t(\mathbf{v})$ or $\mathbf{N}_t^{\mathrm{F}}(\mathbf{f})$ could be an undefined vector which caused by our visibility terms in (10) of every camera returns zero. This may leads to an undefined behavior for our optimization process. Therefore, our framework will need to handle this case, so that at least the surface without information can still be reconstructed with visually appealing output. We define a confidence function $\lambda(\mathbf{v})$ for our observed target normal or can be also called **normal blending weight**. This confidence value decreases as the calculated target normal become unreliable. We then use the confidence term to blend the calculated target normal with smoothed geometric normal using the following equation:

$$\mathbf{N}_t^{\mathrm{blend}}(\mathbf{v}) = \lambda(\mathbf{v})\mathbf{N}_t(\mathbf{v}) + (1 - \lambda(\mathbf{v}))\bar{\mathbf{N}}_g(\mathbf{v}) \tag{17}$$

where, $\bar{\mathbf{N}}_g(v)$ is normal vector of smoothed geometric surface at point $\mathbf{v}$. Our normal blending weight is varied to **the number of visible viewpoints** and **variance of photometric curvature of visible viewpoints**:

$$\lambda(\mathbf{p}) = \lambda_{\mathrm{H}}(\mathbf{p})\lambda_{\mathrm{C}}(\mathbf{p}) \tag{18a}$$

$$\lambda_{\mathrm{H}} = \exp\left(\min\left(0, -\frac{\sigma_{\mathbf{H}}}{2}\right)\right) \tag{18b}$$

$$\lambda_{\mathrm{C}} = \exp\left(\min\left(0, |\mathbf{C}_{\mathrm{seen}}| - \left\lceil \frac{(1-\cos\theta)}{2} \right\rceil |\mathbf{C}|\right)\right) \tag{18c}$$

where λ_{H} is photometric curvature variance term which can be calculated by looking up all normal components from visible cameras and compute its variance. That is, if the photometric normals are consistent, the calculated target normal is more likely to be reliable. Where, λ_{H} captures the photometric curvature variance. The term λ_{C} represents the vertex visibility. If the calculated target normal are computed from more viewpoints, the target normal is acceptable. We assumed that camera set $\mathbf{C}$ are uniformly located along the sphere that covers the scanning object. Then, at a particular vertex $\mathbf{v}$ on

Figure 2: Input reflectance information of **speccat** and **hammerman**. From left to right, diffuse albedo, diffuse reflection, specular albedo, specular reflection, and mask information for our optimization framework.

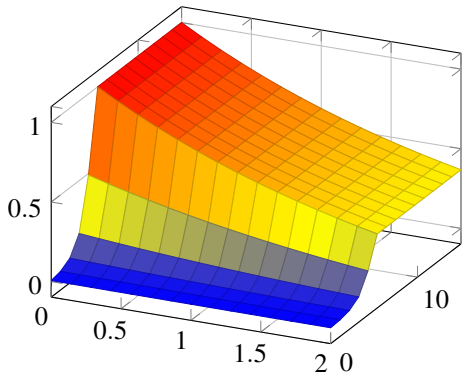


Figure 3: Blending weight function of our camera configuration, varied to $\sigma_{\mathbf{H}}$ and $|\mathbf{C}_{\text{seen}}|$ with 31 total cameras and $\theta = 45°$

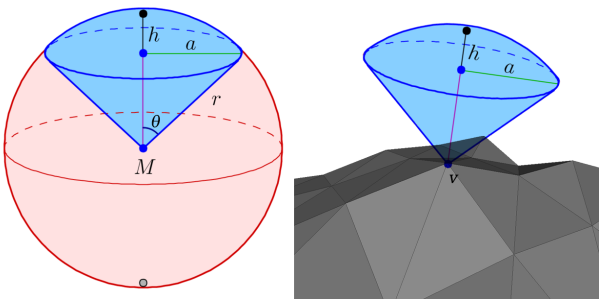


Figure 4: Cameras within an infinite radius spherical sector (hi-lighted in blue) will be marked as visible.

a surface, $\mathbf{v}$ will have a set of visible cameras $\mathbf{C}_{\text{seen}}$. The term $(1-\cos\theta)/2$ is derived from the ratio between surface of spherical sector to the whole sphere, where at a particular point $\mathbf{v}$ should be at least visible to the camera that is located on the part of spherical cap which in this research we set the θ value to be 45 degree. However, this equation is only based on our camera configuration. It could be adjusted to be suitable for other configuration as well.

4.5 Matching Cost Optimization

Normal integration has its limitation about ambiguity as stated in [2], which can result in an incorrect answer even the energy functional is converged. Especially in the concave area where the observed target normal can be inaccurate due to projection error. We can solve such problem by using similar idea to **stereo reconstruction**.

We solved this problem in a similar manner to the normal integration by defining the normal correspondence energy function based on mesh vertices and move them to the optimal solution. Given, a vertex $\mathbf{v} \in \mathbb{R}^3$ and camera set $\mathbf{C}$. We assumed that, if the vertex $\mathbf{v}$ is at the correct position, then its projected normal from each camera should be correspond to every other cameras. In order to find such correspondence, we defined our **matching cost function** to be a variance of observed normal of visible viewpoints where the number of visible viewpoint is more than 3. Otherwise we will force the matching cost to be $+\infty$

We sampled points along the vertex normal both outward and inward, then calculate the matching cost at each sampling. After that, from all computed matching cost samples, we fit a quadratic equation for the displacement from the sample with minimum cost and its adjacent samples. We then compute the location of the tip of the parabola. Finally, based on the matching cost of the optimal point, we translate the vertex along its normal with the displacement calculated earlier.

4.6 Remeshing

The drawback of using explicit surface representation like triangular mesh is that it could not automatically deal with topology changes unlike implicit surface representation. It is likely that self-intersection will occurred in our mesh due to our mesh evolution in Section 4.3.

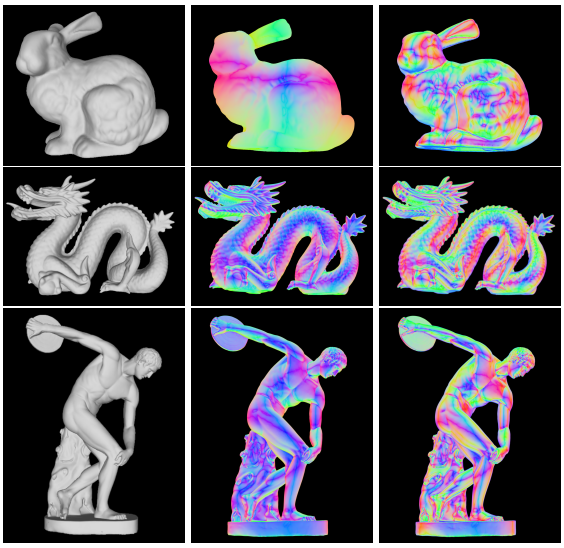


Figure 5: Synthetic data of bunny, dragon, and discolobus. From left to right, ground truth mesh, diffuse reflection, and specular reflection. We omitted diffuse and specular albedo since we set all the value into one.

The author in [20, 21] proposed a framework to efficiently solve topological changes on triangular mesh called **Transformesh**. The algorithm solves topology changes by using an intuitive geometrically driven solution which we found it suitable for our framework. We perform **Transformesh** algorithm after the whole mesh deformation process is completed in every iteration. Other self-intersection algorithm such as [4, 6, 7] can also be used in this step.

Then, after every resolution iteration, we resample our mesh to be finer with edge splitting operation and remove short edge with edge collapsing operation. We then use the mesh in the next optimization iteration.

5 RESULTS

In this section, we will discuss and evaluate the result of our reconstruction pipeline with both real and synthetic data. All procedures are executed with Intel i7-5820K 3.30GHz, with 64GB of RAM.

5.1 Real data

We performed the reconstruction on two real data (As shown in Fig.2). There are 31 viewpoints uniformly located on faces of truncated icosahedral (except one on the bottom-most) with 30 centimeters in radius which we can set our parameter α_r in (2) and (3) to be 30. All input images are captured in 4896 by 3684 pixels and camera matrices are already calibrated. Visual hull is then extracted for initial mesh with 1 millimeter in voxel edge length. We optimized more iterations in coarser resolution mesh as the coarse details will converged before refining the mesh in the finer iteration so that the optimization converges faster in overall. We scheduled 10 iterations or the coarse resolutions, then decreasing the number of optimization iterations in finer resolution iteration.

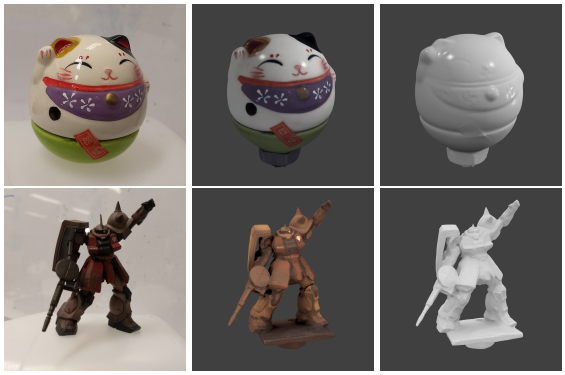


Figure 6: Reconstructed 3D models of speccat and hammerman. The real objects are shown on the left and rendered reconstructed outputs are shown on the middle and right.

Figure 6 shows the results of reconstructed mesh of **speccat** and **hammerman**. Our framework successfully recovered both data. The output of speccat has reasonable geometric features and being able to render an appealing result.

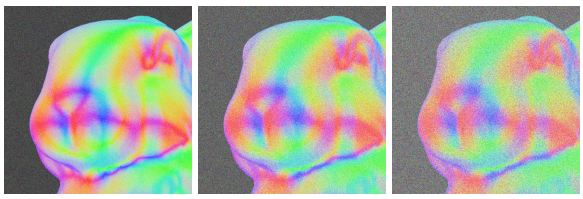


Figure 7: Additive white gaussian noise is added to bunny data with coefficient 0.1, 0.2, and 0.3

5.2 Synthetic data

We simulate the configurations from real data reconstruction from the last section, so that it is not biased or favoring to our framework. We used three ground truth meshes (As shown in Figure 5) and projecting reflectance information needed with similar camera calibration to the prior section. We replaced every pixels to be one for diffuse and specular intensity since there were no such information to project and this would not violate our framework. In addition, **additive white gaussian noise** is added to the generated images with coefficient **AWGN coeff** value set to 0.1, 0.2, and 0.3 (As shown in Fig. 7) and compared the result to evaluate the robustness of our framework.

We measure the error of our output with Hausdorff distance [8] as shown in Table 1. Note that our framework can reasonably recovered the shape even with noisy data. Only the concave area seems to far off the ground truth due to the matching cost optimization could not perfectly find the correct optimal point with noisy data.

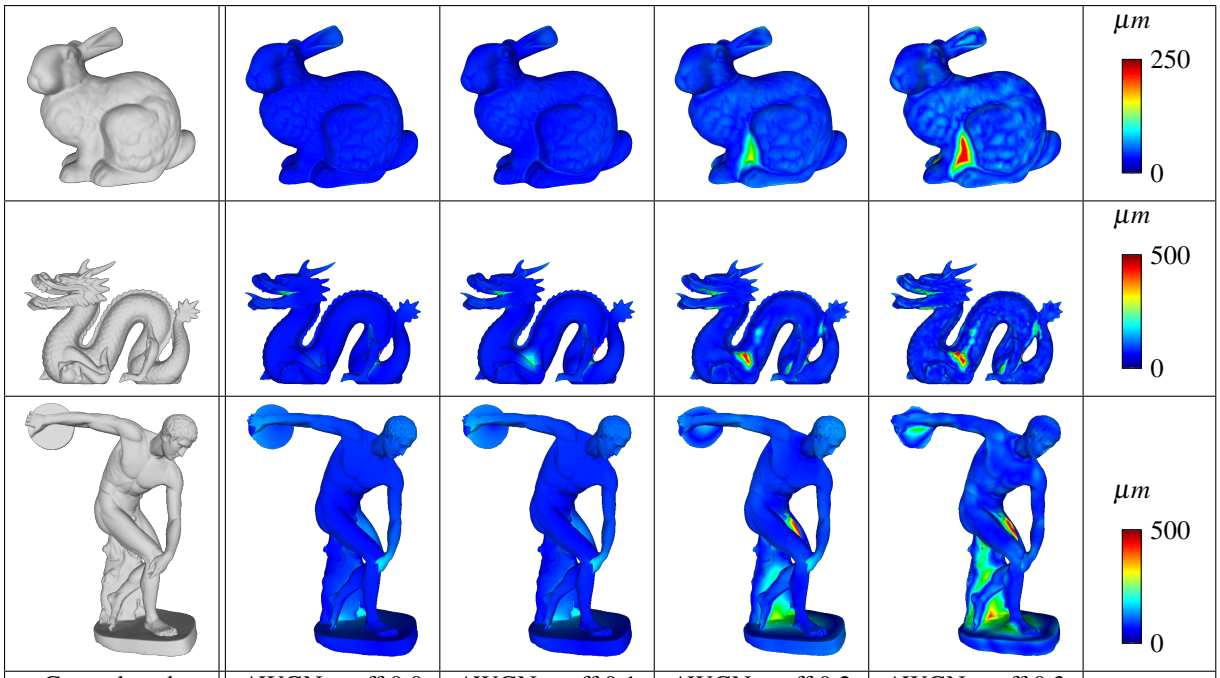


Figure 8: The Hausdorff distance of the outputs toward its ground truth.

Table 1: Hausdorff distance of outputs toward ground truths

Model	AWGN coeff	meand_H (mm)	RMS
Bunny	0	0.06026	0.07269
	0.1	0.06405	0.07791
	0.2	0.16838	0.27443
	0.3	0.23536	0.47291
Dragon	0	0.12251	0.27265
	0.1	0.15447	0.33639
	0.2	0.24416	0.46701
	0.3	0.30370	0.55531
Discolobus	0	0.21390	0.30124
	0.1	0.22146	0.30930
	0.2	0.36786	0.55528
	0.3	0.44007	0.71934

6 CONCLUSION

We have presented a novel multi-view normal integration framework using reflectance information. With our mesh-based optimization, we are able to reconstruct fine details without sacrificing unnecessary memory consumption unlike implicit surface framework. Although it can presents self-intersection, we exploit such problem by using **Transformesh** [20] which fix topology changes completely in triangular mesh domain. We also deal with those surface which only a few or none of the camera can be seen with target normal blending which will smooth out the surface without photometric information.

Our framework also has some limitations toward the area with high details and thin surface. This is due to the inability to observe high frequency target normals on the coarse iterations which result it smoothed out surface like in hammerman (Fig.6). This could be resolved with adaptive mesh w.r.t. photometric curvature and is an interesting area of future work. While our method can handle topological change during optimization, if the initial mesh is of different genus from the real mesh, our algorithm may not be able to change the genus. Hence, using a good initial mesh with matching genus may be needed. Creating a better initial mesh is hence another interesting area of future work.

7 REFERENCES

[1] Sympy. `http://www.sympy.org/`. Accessed: 2018-03-15.

[2] J. Balzer and S. Werling. Principles of shape from specular reflection. *Measurement*, 43(10):1305 – 1317, 2010.

[3] J. T. Barron and J. Malik. Shape, illumination, and reflectance from shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(8):1670–1687, Aug 2015.

[4] G. L. Bernstein and C. Wojtan. Putting holes in holey geometry: Topology change for arbitrary surfaces. *ACM Trans. Graph.*, 32(4):34:1–34:12, July 2013.

[5] J. Y. Chang, K. M. Lee, and S. U. Lee. Multiview normal field integration using level set methods. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[6] N. Chentanez, M. Müller, and M. Macklin. Gpu accelerated grid-free surface tracking. *Computers & Graphics*, 57:1–11, 2016.

[7] N. Chentanez, M. Müller, M. Macklin, and T.-Y. Kim. Fast grid-free surface tracking. *ACM Trans. Graph.*, 34(4):148:1–148:11, July 2015.

[8] P. Cignoni, C. Rocchini, and R. Scopigno. Metro: Measuring error on simplified surfaces. *Computer Graphics Forum*, 17(2):167–174, 1998.

[9] A. Delaunoy and E. Prados. Gradient flows for optimizing triangular mesh-based surfaces: Applications to 3d reconstruction problems dealing with visibility. *International Journal of Computer Vision*, 95(2):100–123, Nov 2011.

[10] C. H. Esteban, G. Vogiatzis, and R. Cipolla. Multiview photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3):548–554, 2008.

[11] R. T. Frankot and R. Chellappa. A method for enforcing integrability in shape from shading algorithms. *IEEE Transactions on pattern analysis and machine intelligence*, 10(4):439–451, 1988.

[12] S. Herbort and C. Wöhler. An introduction to image-based 3d surface reconstruction and a survey of photometric stereo methods. *3D Research*, 2(3):4, 2011.

[13] B. K. Horn and M. J. Brooks. The variational approach to shape from shading. *Computer Vision, Graphics, and Image Processing*, 33(2):174–208, 1986.

[14] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):150–162, Feb 1994.

[15] A. Osep. Multiview normal field integration using graph-cuts. In *Central European Seminar on Computer Graphics for Students*, Apr. 2012.

[16] P. Schroeder, D. Zorin, and W. Sweldens. Subdivision for modeling and animation. 1998.

[17] B. Tunwattanapong, G. Fyffe, P. Graham, J. Busch, X. Yu, A. Ghosh, and P. Debevec. Acquiring reflectance and shape from continuous spherical harmonic illumination. *ACM Transactions on graphics (TOG)*, 32(4):109, 2013.

[18] M. Weinmann, A. Osep, R. Ruiters, and R. Klein. Multi-view normal field integration for 3d reconstruction of mirroring objects. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[19] Y. Yoshiyasu and N. Yamazaki. Topology-adaptive multi-view photometric stereo. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1001–1008. IEEE, 2011.

[20] A. Zaharescu, E. Boyer, and R. Horaud. Transformesh: A topology-adaptive mesh-based approach to surface evolution. In *Proceedings of the 8th Asian Conference on Computer Vision - Volume Part II*, ACCV'07, pages 166–175, Berlin, Heidelberg, 2007. Springer-Verlag.

[21] A. Zaharescu, E. Boyer, and R. Horaud. Topology-adaptive mesh deformation for surface evolution, morphing, and multiview reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4):823–837, 2011.

[22] R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah. Shape-from-shading: a survey. *IEEE transactions on pattern analysis and machine intelligence*, 21(8):690–706, 1999.

Linear Subspaces of the Appearance Space

Mylo, Marlon
Fraunhofer FKIE
Fraunhoferstraße 20
53343 Wachtberg,
Germany
mylo@cs.uni-bonn.de

Klein, Reinhard
University of Bonn
Institute of Computer Science II
Regina-Pacis-Weg 3
53113 Bonn, Germany
rk@cs.uni-bonn.de

Manipulating digital optical material representations is still a difficult problem because arbitrary manipulations lead almost certainly to an unrealistic impression of the material. In this paper we present an approach to material editing based on a digital model of the V1-area of the visual cortex. The V1-model is used to define the appearance space as the space of weighted sums of the cortical-model filter responses. We will show that it is possible to transform several optical material manipulation schemes into our editing scheme. As those optical material manipulation schemes may also be physical phenomena, we may introduce a new material edit. Our argumentation will be supported by comparing editing-examples.

Keywords
Material Editing, BTF, Striate Cortex

1 INTRODUCTION

Editing digital representations of the measured reflectance-properties of material surfaces is an intensely studied but still difficult problem. Renderings of 3D-scenes, which give the impression as if they were real are of high significance e.g. in advertisement, film-productions and historical reconstruction projects. Most approaches target at manipulating the underlying physics whereas we present material editing as a matter of influencing the visual perception. Namely we will transfer several approaches to material editing into a computational model of the simple cells of the **primary visual cortex** (*V1*). Using models of the visual cortex has a long tradition in computer vision for pattern recognition tasks and for the description of perceptual image-metrics but it is not yet an integral component of computer graphics. We will introduce the term **appearance space** which has mostly been used implicitly [15, 24]. We argue, that the set of all possibly occurring neuronal states in the visual cortex may be seen as this appearance space. So given a computational cortex transform model, we may define a *computational* appearance space as part of it. Seeing computer graphics from the perspective of human physiology is fruitful: Bayer filter in digital image sensors follow the cone distribution in the retina, retinal displays mimic the cone-density and photo-sensors filter and cumulate the incoming energy comparable to photo receptor cells. Our approach is a kind of frequency editing. Frequency editing is a very old technique. Blurring e.g. has already been used hundred years ago in silent film. But it has mostly not been seen as an operation in the visual cortex but merely as a given visual effect.

Our point is that optical operations in the physical world are mapped to operations in the appearance space. Recent perceptual studies [11] suggest, that some of those cortical operations are linear. We will show that those operations may be applied by scaling of cortical filter responses. In other words we simulate a physiological linearisation of a physical phenomenon. This is in accordance with results from behavioral and brain science, due to which appearances may be seen as representation of optical phenomena, relevant for the human evolutionary adjustment to our environment [43]. Following this idea, we may state that our visual system is the simplest known representation of optics which still allows all possible perceptual manipulations.

After outlining the relevant related literature we will introduce our model of the V1-cortex. In the third section we will give a formal overview, the fourth section will be dedicated to the description and parametrization of the Gabor-base functions underlying our V1-model. In the fifth section we describe how to transfer optical material manipulation schemes into our model.

The presented results (section 6) will support the conclusion that perceptually motivated frequency manipulations may be seen as promising approach to the generation of new virtual materials (section 7).

https://doi.org/10.24132/JWSCG.2018.26.2.4

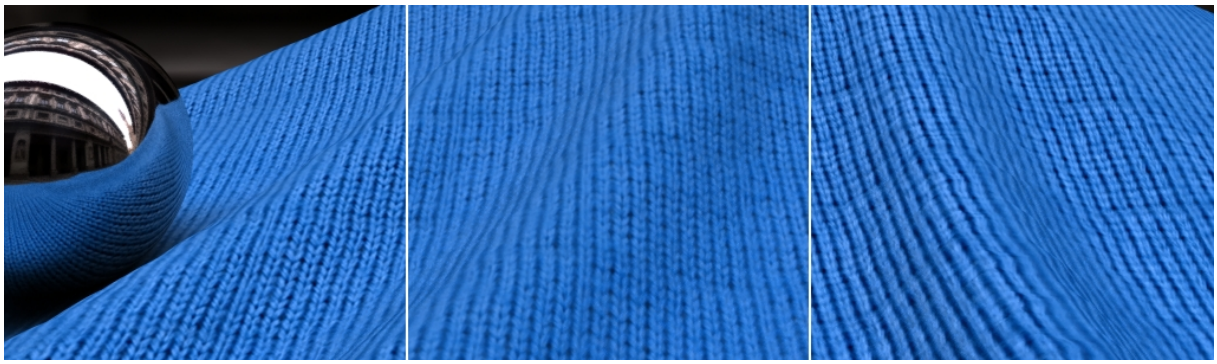


Figure 1: Rendering of an edited wool-BTF. The left part of the image shows the result of a combination of the edge aware operator and the thickening operator (sections 6.2 and 6.3), the right image shows the result of the corresponding band-pass filter, according to [29] and in the middle, we show the original material.

We contribute to the field of material editing by presenting a system to transfer image manipulations into a model of the visual cortex which in many cases brings better results than the original editing scheme and we will provide a novel realistic material manipulation, namely frequency based moving of a light source.

2 RELATED WORK

Because of its outstanding role in visualisation, in advertisement and in filmmaking, editing of realistic material-surfaces is a vivid field of research. In the first paragraph we will portray the development in the field of digital cortex-modelling. The second paragraph will be dedicated to literature on comparable image processing schemes. The related work for the manipulation operators will be presented in section 5.

Our understanding of the structure and the modes of action of the animals visual cortex goes back to the work of Hubel and Wiesel during the late 50. and 60. of the last century [16–18]. Twenty years later Daugman fitted Gaussian and Gabor-filters to the cortical responses measured by Hubel and Wiesel [4, 5]. It is noticeable that neural networks develop Gabor-filter like structures by their own, when trained with random input [41]. Olshausen and Field found that optimizing a vector base for sparse linear coding of images leads to a set of Gabor-like base vectors which is in spatial frequency and orientation coverage comparable to the filtering system in the visual cortex [33, 34]. A publication which concentrates on mathematical aspects of the Gabor-filter-systems compatible with the neural responses of the V1-cortex is the work of Lee [23]. Lee gives explicit parameters for his filter systems and calculates the tightness of the Gabor-frames. A good overview over publications on cortical parameter measurements may be derived from [26, Table 1]. Recently Huth et al. published a python-toolbox for simulations of early vision [19].

The presented approach stands in the tradition of the pyramid-based texture analysis and synthesis published by Heeger and Bergen in 1995 [15]. Heeger and Bergen use steerable pyramids to model the behaviour of the visual cortex. Gutman and Hyvärinen derive a probabilistic model of image statistics by modelling two cortical layers of simple and complex cells [13]. This publication may also be consulted for further references to Bayesian perception. In her dissertation Diana Turcsány [45] uses a convolutional neural network to model the deeper levels of the visual cortex for image editing.

3 OVERVIEW AND DEFINITIONS

The insight that light is not coloured but that the energy in a light beam provokes a sensation of colour goes back to Newton [32]. Heeger and Bergen used the word *appearance* to bridge the gap between the sensation of a texture and the physical phenomena on the surface of the texture [15]. We can locate the term appearance between the sole occurrence of physical phenomena and the set of sensations by identifying the **appearance space** with the set of all neural response-states in the visual cortex. In the ventral stream of the human visual system, the primary visual cortex follows after the **lateral geniculate nucleus** (*LGN*). As receptive fields have directly been measured while exposing the macaque retina to visual stimuli, the influence of the LGN is an implicit part of the model but does not have to be modelled explicitly.

As frame for our (computational) appearance space we will use a **cortex transform model** [46] which we will derive from empirical data (section 4).

Our formal scaffold consists of a **model of the space of retinal responses**, a **model of the neural responses of simple cells in V1**, a **model of the visual stream** from the retina to the neural response and an **interpretation model for the retinal responses**.

The space of retinal responses describes the entrance of pictorial data into the visual system. We will use RGB-images with an edge-length of 256 pixels. Decorrelating the color space as in [15, Sec. 3.5], lead to

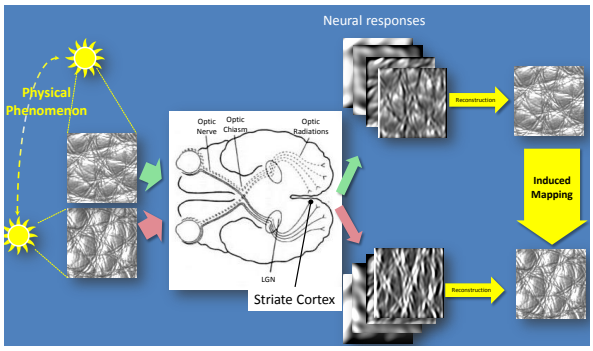


Figure 2: Visual path of a material patch, seen under different optical conditions. The physical phenomenon induces a mapping in the space of cortical responses. Brain drawing taken from http://universe-review.ca/I10-85-opticpath.jpg.

strong artefacts. Confining the manipulations to the value-channel of the HSV-color-space brought good results. So we define $\mathbb{I} := [0,1]^{256\times256}$ as retina model. The images, we use for testing, correspond to real-world patches with an edge length of approximately 5 cm. If a patch of this size has a distance of 57 cm from the observer, its retinal image approximately covers the fovea.

Our model of the visual stream is limited to the early ventral stream up to V1. While there have been suggested different filters for modelling V1-receptive fields [25], we use Gabor-filters [4], (see section 4). Our whole V1-model consists of a filter bank of 517 filters ($\{\Gamma_\psi\}_{\psi\in\Psi}$, see section 4.2.1).

The space of neural responses will be modelled as a stack of matrices $\mathbb{G} := \mathbb{R}^{256\times256\times517}$. We do not limit the amplitude of neural responses. It is not self-telling, that the spatial dimension of the neural responses (256×256) equals the dimension of $\mathbb{I}$ (see paragraph 4.2.2) but it enables a direct comparison between the input and the result of the V1-transform.

The interpretation space is a set of mappings $\mathbf{G} : \mathbb{G}\to\mathbb{I}$ with $\mathbf{G} := \sum_\psi a_\psi\{\Gamma_\psi \star \mathcal{T}\}$.

Now we define the **appearance space** $\mathbb{A}$ as the image of the interpretation space. This leads to the following diagram, modelling the relations, depicted in figure 2:

$$\begin{array}{ccccc} \mathcal{T}_{\in\mathbb{I}} & \xrightarrow{\{\Gamma_\psi\}} & \{\Gamma_\psi\star\mathcal{T}\}_{\in\mathbb{G}} & \xrightarrow{\mathbf{G_I}} & \mathbf{G_I}\mathcal{T}_{\in\mathbb{A}} \\ \downarrow\mathfrak{P} & & & \searrow^{\mathbf{G}_\mathfrak{P}} & \downarrow\mathbf{E}_\mathfrak{P} \\ \mathfrak{P}(\mathcal{T})_{\in\mathbb{I}} & \xrightarrow{\{\Gamma_\psi\}} & \{\Gamma_\psi\star\mathfrak{P}(\mathcal{T})\}_{\in\mathbb{G}} & \xrightarrow{\mathbf{G_I}} & \mathbf{G_I}\mathfrak{P}(\mathcal{T})_{\in\mathbb{A}} \end{array} \tag{1}$$

The filter bank $\{\Gamma_\psi\}$ maps the texture $\mathcal{T}$ to the neural response space $\mathbb{G}$. By the definition of $\mathbb{G}$ and $\mathbb{A}$, we may identify $\mathbb{I}$ and $\mathbb{A}$. Neural responses are recombined to a texture $\mathcal{T}_\mathcal{X} := \mathbf{G_X}\mathcal{T}$ in the appearance space. A physical phenomenon $\mathfrak{P}$ induces a mapping from the appearance space $\mathbf{E}_\mathfrak{P} : \mathbb{A}\to\mathbb{A}$ to itself (compare with figure 2). If we identify $\mathbf{G_X}$ and $\mathbf{G_X}\circ\{\Gamma_\psi\}$, the operator $\mathbf{G}_\mathfrak{P}$ may be constructed as linear approximation of $\mathbf{E}_\mathfrak{P}$.

$$\mathbf{G}_\mathfrak{P} \approx \mathbf{E}_\mathfrak{P} \tag{2}$$

Note that $\mathbf{G_I}\approx\mathbf{I}$ is an approximation of the identity on $\mathbb{A}$. $\mathbf{G}_\mathfrak{P}$ is the operator, we want to learn. For a full clarification of the symbols, see the following section.

4 THE COMPUTATIONAL MODEL OF THE EARLY VISION

In this section we will introduce the V1-model. The concept that the neural response of a simple cell in V1 cortex is linear in the intensity of the incoming optical stimulus is essential not only for the model of the visual pathway [1,4] but also for all measuring methods of the receptive fields like subspace reverse correlation [39]. The function describing the weighted contribution from each position of the receptive field to the response of this cell is called **weighting function** and may be modelled by a linear filter [46].

4.1 An empirically based model of the visual cortex

There exist many publications on the frequency distribution in Macaque V1-area [9, 42]. We used empirical data, measured and fitted by De Valois et al. [6]. We use two dimensional Gabor-base functions for spatial frequency filtering [4]. It is convenient, to introduce the Gabor-filtering system by starting with a transformation of the euclidean plane:

$$R_\Theta\circ T_\mathbf{p}(x,y) = \begin{pmatrix}\cos\Theta & \sin\Theta\\ -\sin\Theta & \cos\Theta\end{pmatrix}\begin{pmatrix}x-p_x\\ y-p_y\end{pmatrix} \tag{3}$$

With the point $\mathbf{p} := \begin{pmatrix}p_x\\ p_y\end{pmatrix}$ and the rotation angle Θ. The Gabor base function is the product of a wave-function, called **carrier** (cos), and an Gaussian **envelope** (exp):

$$a\gamma_{\omega,\sigma_\xi,\sigma_\eta,\phi}(\xi,\eta) = ae^{-(\xi/\sqrt{2}\sigma_\xi)^2-(\eta/\sqrt{2}\sigma_\eta)^2}\cos(\omega\xi+\phi) \tag{4}$$

Here we use ξ and η for the position to emphasize that it refers to the local coordinate system. The preimage of the directional standard deviation of the Gaussian envelope forms an ellipse. The semi-minor axis, here the ξ-axis, of this ellipse is according to [23] and [20, Fig. 8A] parallel to the wave-vector of the carrier. We confine to a real plane-wave (see 4.2.1). So, with $\psi := \{\omega,\sigma_\xi,\sigma_\eta,\phi,\Theta\}$, we may define:

$$a\Gamma_{\psi,\mathbf{p}} := a\gamma_{\omega,\sigma_\xi,\sigma_\eta,\phi}\circ R_\Theta\circ T_\mathbf{p}(x,y) \tag{5}$$

4.2 Parameters

To compose the Gabor filter bank, we have to specify the parameters. We distinguish between the parameters, which we set up according to given publications in the field of neuro-science (the parameter set ψ, paragraph 4.2.1), the position of the filter center $\mathbf{p}$ (paragraph 4.2.2) and the amplitude a (paragraph 4.2.3), which we will use for the definition of the editing operator $\mathbf{G}$.

4.2.1 The parameter-set ψ

The parameter-set ψ contains all parameters which have to be distributed according to measurements in the macaques or in the cats striate cortex.

The spatial frequency ω

In the visual cortex, frequency sensitivity occurs not in exact but in rough steps of 0.3 to 0.5 octaves. As we drew the spatial frequency according to [6, Fig. 6.], we were limited to the bin width in this figure, which is 0.5 octaves.

Differences between human and macaque visual system The monkey visual system as model for the human visual system has been validated under several different aspects [37]. While the human visual system is from an anatomical and physiological perspective extremely similar to the macaque visual system, it has a slightly higher retinal magnification factor (about $0.291/0.223$), which hints to a higher angular resolution [28]. Therefore we add another frequency bin at $20.8\text{c}/^\circ$ and so we have to extrapolate to a plausible histogram of the human frequency distribution.

The distribution given by de Valois In [6, Fig. 6.] De Valois et al. describe their measurements of the spatial frequency distribution of the receptive fields of simple cells in macaques primary visual cortex. They distinguish between the cells with receptive fields in the fovea and in the parafovea region of the retina. We assume that our texture covers a visual angle of 5°. As we cannot expect observers to concentrate on a texture without any eye-movement, we merged the distributions for the fovea and the parafovea by normalized summation. The blue (including green-blue) bars in Figure 3 belong to the merged histogram from [6]. To extend this histogram to the slightly bigger frequency range of the human vision, we fitted a gaussian by an iterative Least Mean Square algorithm, moved the mean of the gaussian to the logarithmic middle of the new frequency distribution range and stretched the standard deviation proportional to the ratio of logarithmic ranges.

The following table shows the number of filters we have in every frequency bin. The absolute number of 517 filters has been chosen in order to have a good fitting to the histogram and still stay comparable with the reconstruction scheme for tight frames (section 6.1).

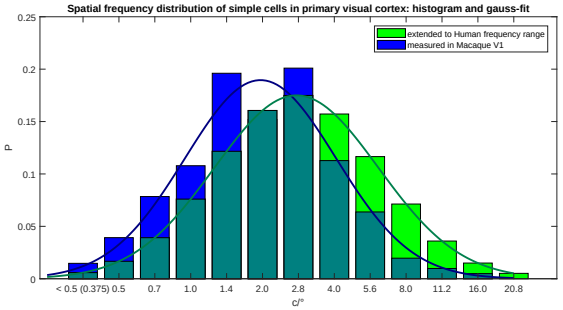


Figure 3: The blue part of the bars shows the histogram given in [6]. The green part describes the extrapolation results and has been added to account for the slightly wider frequency range of human vision [28].

c/°	0.4	0.5	0.7	1.0	1.4	2.0	2.8
# filters	3	8	20	39	64	84	92
c/°	4.0	5.6	8.0	11.2	16.0	20.8	
# filters	82	61	37	18	7	2	

The standard deviation in direction of the wave-vector σ_ξ

σ_ξ and ω are connected via the bandwidth. As Gaussian kernels have infinite support, the bandwidth is defined as **half amplitude bandwidth**. Bandwidths have been drawn on base of [6, Fig. 7]. In this diagram, De Valois et al. visualized the bandwidth with standard deviation as a function of the spatial frequency. As spatial frequencies were known, bandwidth-samples could be drawn under the assumption of normal-distribution within the same frequency range. Given the bandwidth B and the spatial frequency ω, we may calculate:

$$\sigma_\xi = \frac{\sqrt{2\ln 2}\left((2^B+1)/(2^B-1)\right)}{\omega} \tag{6}$$

The standard deviation orthogonal to the wave-vector σ_η

According to [38, FIG. 4.], there is a relation between $\omega\sigma_\xi$ and $\omega\sigma_\eta$. This relation may be interpreted as functional graph with a small deviation. To make use of this relation, we fitted a cubic spline to the data and used this spline as function graph.

The Phase angle ϕ

To draw the phase parameter ϕ, we used the histograms given in [38, FIG. 7A/B].

The orientation Θ

By definition of $\xi = (x - p_x)\cos\Theta + (y - p_y)\sin\Theta$, Θ is the angle between the ξ and the x-axis. We drew the orientation equally distributed from $\{i\frac{\pi}{8}\}_{i\in\{1,\dots,16\}}$. Where possible, directions have been drawn in orthogonal pairs.

All random experiments have been done in several passes and brought comparable results. The set of all parameter-sets ψ in the Gabor-filter bank, will be denoted by Ψ.

4.2.2 The position $\mathbf{p}$

Every neural measurement provides us just one sample of the domain of neural responses. Be Γ the Gabor filter, best fitting the receptive field of a given neuron with filter center $\mathbf{p}$: now the neural response is modelled as $a\langle\Gamma_{\mathbf{p}}, \mathcal{T}\rangle$.

$\langle\rangle$ is the standard inner product in the image domain. Γ has to be appropriately sized and evaluated on the spatial grid of the image and $\mathcal{T}$ has to be zero-padded, where necessary. The filter-centers are often chosen to be elements of the spatial image grid ($\mathbf{p} \in \{1,\dots,256\}^2$) [5], sometimes with the constant stride ($h := p_{z_{i+1}} - p_{z_i}$) between consecutive grid points increasing with an increasing wavelength and/or starting with a stride smaller than one (e.g. [23]). In order to make use of the convolution theorem and to avoid a resampling step we will assume the parameter set ψ to be constant over the whole grid and set the stride $h = 1$ to one and keep the image-grid. Nevertheless we have to emphasize that our approach might distract the statistics: as the statistics of DeValois et al. [6] are based on the measurements of individual cells, a higher spatial resolution goes to the cost of the angular resolution and the variety of the phase values. Particularly in the case of low frequencies, the spatial domain is probably oversampled. The results of the undulation experiment 6.3.2 might indicate this problem (see figure 11).

To locate the neural response, we multiply it by the canonical base matrix $\mathbf{e}_{\mathbf{p}} \in \mathbb{R}^{256\times256}$ at position $\mathbf{p}$ and sum those matrices up $\sum_{\mathbf{p}}\langle\Gamma_{\mathbf{p}}, \mathcal{T}\rangle\mathbf{e}_{\mathbf{p}}$. As we confined to real-valued Gabor-base functions (equation 4), meaning $\Gamma^* = \Gamma$, we may write that summation-formula as cross-correlation $\star$:

$$a\sum_{\mathbf{p}}\langle\Gamma_{\mathbf{p}}, \mathcal{T}\rangle e_{\mathbf{p}} = a\Gamma \star \mathcal{T} \tag{7}$$

Where we appoint the amplitude a to be fixed for a changing position $\mathbf{p}$.

4.2.3 The amplitude a

We use the amplitude to combine filter responses to operators. Be $\mathbf{E} : \mathbb{A} \to \mathbb{A}$ an operator, than we want to find $a^{\mathbf{E}}_{\psi}$ to approximate $\mathbf{E}$ (see section 5.1):

$$\mathbf{E} \approx \sum_{\psi\in\Psi} a^{\mathbf{E}}_{\psi}\Gamma_{\psi}\star \tag{8}$$

This mapping operates via cross-correlation, it may be visualized by applying it to the discrete dirac $\delta \in \mathbb{A}$.

5 TRANSFERRING EDITS TO THE MODEL OF THE VISUAL CORTEX

Now, that we have introduced our model of the visual cortex, we want to introduce the operators. First we will discuss the editing scheme itself and how to transform into it. Than we will present the editing paradigms to transfer.

5.1 Learning an operator

To transfer a given edit, we take a collection of test-textures $\mathcal{T}_{i\in\{1,\dots,m\}}$ and solve

$$\mathbf{a}^{\mathbf{E}}_i = \min\arg_{\mathbf{c}\in\mathbb{R}^n} ||\mathbf{E}\mathcal{T}_i - \sum_{\psi\in\Psi} c_{\psi}\Gamma_{\psi}\star\mathcal{T}_i||_2 \tag{9}$$

for each texture $\mathcal{T}_i$. We could define $\mathbf{a}^{\mathbf{E}} := \min\arg_{\mathbf{c}\in\mathbb{R}^n}\sum_i ||\mathbf{E}\mathcal{T}_i - \sum_{\psi} c_{\psi}\Gamma_{\psi} \star \mathcal{T}||_2$ but this definition lead to undesired activities in higher frequency-bands. Instead, we apply a singular value decomposition to $\mathbf{A} := a_{i\psi}$:

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}' \tag{10}$$

and use the base vector $\mathbf{a}^{\mathbf{E}} = (a^{\mathbf{E}}_{\psi})_{\psi\in\Psi} = (V_{\psi 1})_{\psi\in\Psi}$. So we may declare our new editing operator

$$\mathbf{G}_{\mathbf{E}} := \sum_{\psi\in\Psi} a^{\mathbf{E}}_{\psi}\Gamma_{\psi}\star \tag{11}$$

5.2 The operators

We will explore four different operators: the identity, linear edge enhancement, bandpass filters and spotlight moving.

5.2.1 The identity

The first operator maps the image to itself. This is a reconstruction. There is no canonical reconstruction scheme for Gabor-Wavelets as they are overcomplete. There has been many efforts to produce models of the visual cortex which had good mathematical properties [23, 40]. Lee introduces a reconstruction scheme which relies on the tightness of the frame [7, 23]. There are many approaches, to adjust the filter responses of a Gabor-filter bank to a partition of unity in the frequency domain [46].

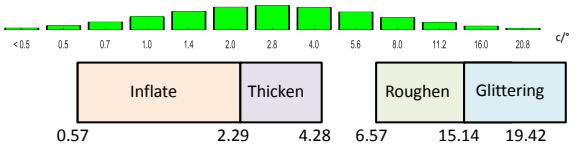


Figure 4: Under the frequency distribution, we show the range of the edits of the editing scheme.

5.2.2 *Edge aware imaging*

Edge aware imaging has been subject of intensive study during the last years. Bilateral filters [36,44] are among the most popular tools for edge-aware image processing. One recent approach gives a linear approximation of a bilateral filter [30]. He et al. suggest to improve the edge-preservation property of filters by the use of a guidance map [14]. Paris et al. use Laplacian Pyramids for strengthening or weakening edges in images: they argue, that edges are a jump in not only one level in the laplacian pyramid but merely in all levels [35]. Laparra et al. use those insides to build a system for perceptually optimized image rendering [22]. Fattal [8] detects edges by the use of second generation wavelets. Using Gabor-filters for edge detection has a long tradition, e.g. [27].

5.2.3 *Bandpass filtering*

Affordance is a concept from psychology, introduced by Gibson [10], and describes the possibilities of actions which may be done on a given object. By user studies, Giesel and Zaidi found a relation between certain affordances or material properties and spatial frequency bands in material-images [11]:

0.57-2.29	c/°	Inflated and deflated
2.29-4.28	c/°	Deep and flat
6.57-15.14	c/°	Soft and rough
15.14-19.42	c/°	Sparkling and dull

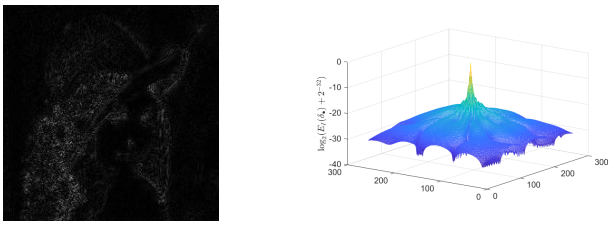
The connection between affordance and spatial-frequencies gives rise to a semantically founded editing scheme by simply enhancing or weakening particular frequency bands [12, 29]. Because the underlying physical effects are too complicated, those effects may not be seen as the result of inverse optics [12] and are therefore examples for complex physical operations with a linear representation in the visual cortex. It is striking, that those manipulations cover nearly the whole frequency range of the visual cortex (figure 4).

5.2.4 *Moving spotlight*

Given a directionally illuminated texture patch. We will show, that it is possible in our model to learn and reproduce small movements of the light source.

6 RESULTS

In this section we want to show and discuss some results. The presented results have been calculated on colour or reflectance maps. The colour maps had a dynamic of 48 dB and the reflectance maps had a dynamic of 96 dB. All operations on the HDR-images had been performed in log-space. As training samples, we used textures from the USC-SIPI Image Database from the University of Southern California and the describable texture dataset [2].



(a) $500 \cdot |\mathcal{L} - \mathbf{E_I}(\mathcal{L})|$ (b) $\log(\mathbf{E_I}(\delta) + 2^{-32})$

Figure 5: 500 times amplified reconstruction error of Lena image. Boundary cut off in a distance of 10 pixel. (a). To get a better impression of the delta-spike, we added 2^{-32} and applied the binary logarithm (b).

6.1 Identity

Figure 5b visualizes the learned identity operator. The difference between the reconstructed and the original image is with bare eyes intractable. The maximum pixel intensity difference between the Lena image ($\mathcal{L}$) and the reconstruction of it was $\max_{ij} |\mathcal{L}_{ij} - \mathbf{G_I}(\mathcal{L})_{ij}| = 6.9 \cdot 10^{-3}$ which corresponds to 2 steps in an 8 bit grey-scale image. While such a small deviation will not stand-out when affecting the intensity channel, sensible people might perceive colour aberrations if the operator was applied channel-wise to an RGB-image.

As it is not possible, to reconstruct an Gabor-filtered image perfectly, we will compare against the approximative reconstruction scheme, presented by Lee [23]: a frame $\{\Gamma_\beta\}$ (for a definition and constraints on the parameter set β, see [7, 23]) is tight when the following equation holds for a given constant c and a small positive number ε:

$$\forall \mathcal{T}: \qquad c||\mathcal{T}||^2 \leq \sum_\beta |\langle \Gamma_\beta, \mathcal{T} \rangle| \leq (c+\varepsilon)||\mathcal{T}||^2 \quad (12)$$

Lee investigated for which parameter sets β this frame becomes a tight frame ($\varepsilon \searrow 0$). Note that Lee uses complex-valued Gabor-base functions, which does not make sense in our setting as we do not apply filters to filtered values and have therefore no complex multiplications. In his definition of the Gabor-base functions, the amplitude a is part of the definition of Γ and the position $\mathbf{p}$ is an element of the parameter set β and he uses a pyramid sampling scheme. For a tight frame the following reconstruction formula may be applied:

$$\mathcal{T} \approx \frac{2}{2c+\varepsilon} \sum_\beta \langle \mathcal{T}, \Gamma_\beta \rangle \Gamma_\beta \quad (13)$$

To compare against [23], we sample over 16 directions θ, made three steps per octave and set the stride h to 0.5. This yields a value for ε of approximately 0.0001, the number of base-vectors was 864.

Figure 6 visualizes that even with this very tight frame the quality of this reconstruction scheme is not high enough to allow for applications in computer graphics.

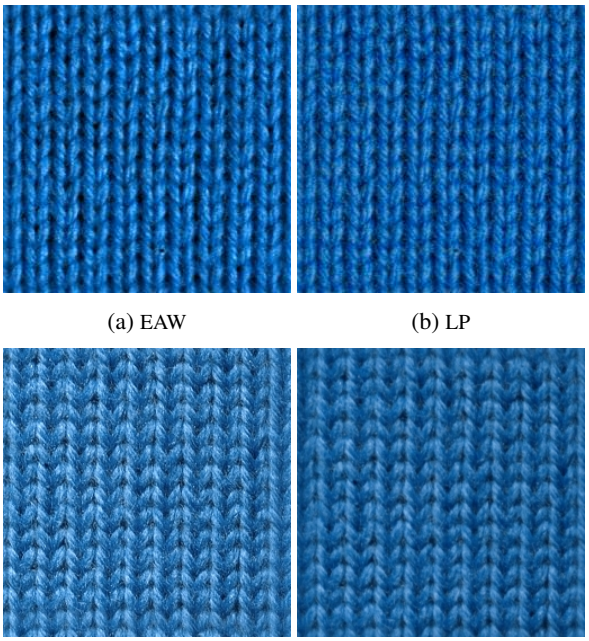
Figure 6: The upper right of the image shows the Lena image reconstructed by the formula 13. The wavelet family forms a frame with $\delta = 0.0001$. The lower left shows our reconstruction.

6.2 Edge aware imaging

For this edit, we learned randomly linear edge aware filtering kernels: we used Gabor and Sobel-filters $((1,2,1)' \otimes (1,0,-1)$, we will write: **SX**,**SY**). Note that the parameters of the filter kernels and the intensity of the filters had been drawn randomly and so they were in general not in the set Γ_Ψ. Intensities were always enhanced. We used 1000 editing samples of varying photos for learning. The resulting filter (figure 8) may be seen as the average of all projected filter-kernels. It is a good approximation of the sign-inverted discrete Laplace operator with weights on the diagonals. In comparison with other state of the art edge aware imaging operators (figure 7), it is noteworthy, that the learned operator enhances very fine structure and the material still looks realistic. A physical effect, bound to this appearance, is a higher fibrousness.

6.3 Affordance editing

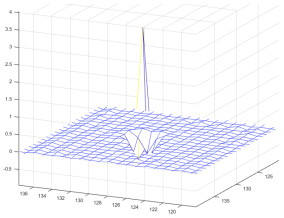
In this section we will compare our results against pure frequency band scaling. While there is evidence, that the frequency-bands are subject to a recognition step [12] and consecutively to a scaling step in the visual cortex, according to the original perceptional studies, an edge length of a material patch should cover a viewing angle of 3.5°, this corresponds to an observer-distance of approximately 82 cm. We will confine to the roughen and the undulation operation. The thicken and the glitter-operator will be compared on bidirectional texture functions (section 6.5).



(a) EAW (b) LP

(c) $\mathbf{G}_{\mathbf{SX},\mathbf{SY},\Gamma}$ (Our) (d) Original

Figure 7: In the top row you can see as comparison the results of two non-linear edge filters: the edge avoiding wavelets of Fattal et al. (7a, [8]) with an exponent of 1.15 (slightly enhancing fine details, see publication) and the local Laplacian filters of Paris et al. (7b, [35]) with $\sigma_{\text{publ}} = 0.2$ and $\alpha_{\text{publ}} = 0.2$. The bottom row shows the result of our algorithm (7c) and the original material (7d).



(a) Mesh

0,009	0,033	-0,017	0,040	-0,001
0,054	-0,103	-0,436	-0,123	0,067
0,012	-0,551	3,155	-0,568	0,038
0,072	-0,120	-0,462	-0,090	0,049
0,000	0,053	0,008	0,012	0,023

(b) Values

Figure 8: The learned filter.

The comparison edit

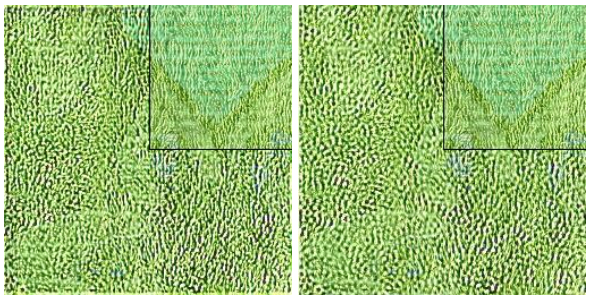
The influence of the absolute value for the strength of the edit is not directly comparable. The learned edits were mostly weaker than the originals. To compensate for that, we made a relaxation step based on the HDR VDP 2.2-metric as published by Mantiuk et al. in 2015 [31], by scaling the edit with a positive number s with

$$s := \text{argmin}_{r>0} \left| \text{d}(\mathcal{T}, \mathbf{E}\mathcal{T})_{VDP} - \text{d}(\mathcal{T}, r\mathbf{G}_\mathbf{E}\mathcal{T})_{VDP} \right| \tag{14}$$

to minimize the visual difference between $\mathbf{E}$ and $\mathbf{G}_\mathbf{E}$. Of course $r\mathbf{G}_\mathbf{E} := r(\mathbf{G}_\mathbf{E} - \mathbf{G}_\mathbf{I}) + \mathbf{G}_\mathbf{I}$

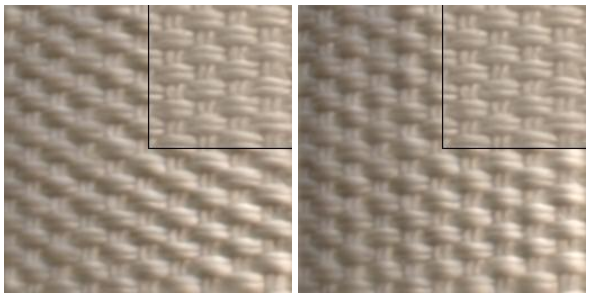
6.3.1 Roughening

Roughening seems to work comparably good in the Fourier-domain (operator $\mathbf{F}$) and in the Cortex-filter-



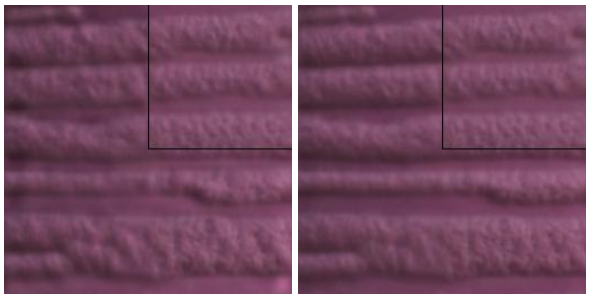
(a) $\mathbf{G}_6$ (our) (b) $\mathbf{F}_6$

Figure 9: Comparison of the roughening filter. The original material in the top right corner.



(a) **G** (our) (b) **F**

Figure 10: Inflating a material. The original material in the top right corner.



(a) **G** (our) (b) **F**

Figure 11: Inflating a material. The original material in the top right corner.

bank. For stronger edits (images 9a and 9b) our operator shows less artefacts.

6.3.2 Inflation

For relatively small structures, the undulation-operation works slightly better in the cortical filter bank (image 10a) than in the fourier domain (image 10b, operator **F**). For bigger structures, the manipulation in the cortical filter bank is not capable of reproducing the results of the bandpass filtering in the Fourier-domain (figure 11).

6.4 Spotlight moving

To make the moving spotlight experiment, we used the BTF-measurements of the UBO14-database of the university of Bonn [47]. For learning, we used the leather

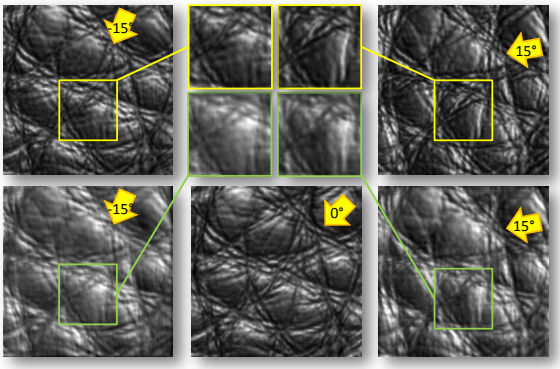


Figure 12: In the middle of the bottom row, you can see the original material test-patch. To compare against the real physical operation, we show in the top row a photography of the same patch, illuminated under an azimuthal angle of $-15°$ (left) and illuminated under an azimuthal angle of $15°$ (right). We compare those results against the application of the spotlight moving-operator (bottom row, left ($-15°$) and right ($15°$)). Here we show only the value channel of the material patch.

materials with the numbers 1-3 and 5-12. The testing results will be presented on the leather4 material. The camera position had been in the zenith above the material. Material-patches which were illuminated from a polar angle of approximately $30°$ against the zenith and from an azimuthal angle of $0°$ were considered as unedited material samples. We interpreted material patches, taken under the same conditions but illuminated from an azimuthal angle of $15°$ or $-15°$ respectively as the edited versions of the original material patch and used those patches for learning the motion of the spotlight. The results are presented in figure 12 and in a short movie in the additional material.

We can see that small moves of a spotlight can be represented and learned in the cortical domain.

6.5 Editing of high-dimensional material representations

Image based modelling of spatially varying and measured material reflectance properties has been introduced by Dana et al. [3] in form of bidirectional texture functions (**BTF**s). An approach to editing BTFs is, to deal with BTFs as with textures. In 2007, Kautz et al. showed, that applying operators from picture editing to the spatial or to the angular domain of a BTF may bring reasonable results [21] and introduced several different operators to BTF-editing. Our BTF editing approach is comparable to the frequency band scaling, published by Mylo et al. [29]. A detailed description of editing compressed BTF-data may be found there. In figure 1 we compare against the thicken and the glittering-operator from the band scaling approach ([29], see sec-

tion 5.2.3). Here the results of our V1-editing approach are clearly superior.

Energy preservation and other expressions of physical phenomena are lost after the editing step. Instead of using the suggested thicken or the undulation-operator, one may estimate the surface structure and operate on the new geometry. For higher frequencies, the inverse optics are highly complicated.

6.6 Time requirement

Experiments have been done on an *i7-4770 CPU @ 3.4 GHz* with 8 GB RAM. Learning took between $51''$ (spotlight moving, 6.4) and $90'$ (edge aware imaging, 6.2). The editing step took about $0.4'$ for a texture and $40'$ for a BTF.

7 CONCLUSION

In this work we have presented different linear editing operations based on a model of the V1-region of the visual cortex. We could show, that it is possible to reconstruct material patches in an appropriate quality by simple summation of the filter responses of the suggested Gabor-filter bank (section 6.1). We learned band-pass filtering which shows in many cases less artefacts when the corresponding band-pass filter in the fourier domain (section 6.3). This effect might be due to the immanent pre-filtering but we have to pronounce that prefiltering the bandpass in the fourier-domain is difficult because it varies between destroying the effect and producing strong sidelobes. Applying those learned operators to a BTF brought notably better results than the corresponding band-pass filters (section 6.5). Above that, we could also learn a physical effect (section 6.4).

An important subset of the appearance space is the set of all realistic appearances, meaning appearances which are inter-subjectively considered as pictorial representations of a real environment like e.g. photos. We have shown, that starting with a valid element of the space of realistic appearances, the presented operators define an affine linear subspace with limited diameter.

8 REFERENCES

[1] B. Andrews and D. Pollen. Relationship between spatial frequency selectivity and receptive field profile of simple cells. *The Journal of physiology*, 287(1):163–176, 1979.

[2] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[3] K. J. Dana, B. van Ginneken, S. K. Nayar, and J. J. Koenderink. Reflectance and texture of real-world surfaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 151–157, 1997.

[4] J. G. Daugman. Two-dimensional spectral analysis of cortical receptive field profiles. *Vision research*, 20(10):847–856, 1980.

[5] J. G. Daugman. Complete discrete 2-d gabor transforms by neural networks for image analysis and compression. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(7):1169–1179, 1988.

[6] R. L. De Valois, D. G. Albrecht, and L. G. Thorell. Spatial frequency selectivity of cells in macaque visual cortex. *Vision research*, 22(5):545–559, 1982.

[7] R. J. Duffin and A. C. Schaeffer. A class of nonharmonic fourier series. *Transactions of the American Mathematical Society*, 72(2):341–366, 1952.

[8] R. Fattal. Edge-avoiding wavelets and their applications. *ACM Transactions on Graphics (TOG)*, 28(3):22, 2009.

[9] K. Foster, J. P. Gaska, M. Nagler, and D. Pollen. Spatial and temporal frequency selectivity of neurones in visual cortical areas v1 and v2 of the macaque monkey. *The Journal of physiology*, 365(1):331–363, 1985.

[10] J. J. Gibson. Perceiving, acting, and knowing: Toward an ecological psychology. *The Theory of Affordances*, pages 67–82, 1977.

[11] M. Giesel and Q. Zaidi. Adaptation reveals frequency band based inferences of material properties. volume (In press), 2012.

[12] M. Giesel and Q. Zaidi. Frequency-based heuristics for material perception. *Journal of vision*, 13(14):7–7, 2013.

[13] M. U. Gutmann and A. Hyvärinen. A three-layer model of natural image statistics. *Journal of Physiology-Paris*, 107(5):369–398, 2013.

[14] K. He, J. Sun, and X. Tang. Guided image filtering. *IEEE transactions on pattern analysis and machine intelligence*, 35(6):1397–1409, 2013.

[15] D. J. Heeger and J. R. Bergen. Pyramid-based texture analysis/synthesis. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 229–238. ACM, 1995.

[16] D. H. Hubel and T. N. Wiesel. Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 148(3):574–591, 1959.

[17] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106–154, 1962.

[18] D. H. Hubel and T. N. Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, 195(1):215–243, 1968.

[19] J. Huth, T. Masquelier, and A. Arleo. Convis: A toolbox to fit and simulate filter-based models of early visual processing. *bioRxiv*, page 169284, 2017.

[20] J. P. Jones and L. A. Palmer. The two-dimensional spatial structure of simple receptive fields in cat striate cortex. *Journal of neurophysiology*, 58(6):1187–1211, 1987.

[21] J. Kautz, S. Boulos, and F. Durand. Interactive editing and modeling of bidirectional texture functions. In *ACM Transactions on Graphics (TOG)*, volume 26, page 53. ACM, 2007.

[22] V. Laparra, A. Berardino, J. Ballé, and E. P. Simoncelli. Perceptually optimized image rendering. *arXiv preprint arXiv:1701.06641*, 2017.

[23] T. S. Lee. Image representation using 2d gabor wavelets. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18:959–971, 1996.

[24] S. Lefebvre and H. Hoppe. Appearance-space texture synthesis. *ACM Transactions on Graphics (TOG)*, 25(3):541–548, 2006.

[25] T. Lindeberg. A computational theory of visual receptive fields. *Biological cybernetics*, 107(6):589–635, 2013.

[26] F. Mechler and D. L. Ringach. On the classification of simple and complex cells. *Vision research*, 42(8):1017–1033, 2002.

[27] R. Mehrotra, K. R. Namuduri, and N. Ranganathan. Gabor filter-based edge detection. *Pattern recognition*, 25(12):1479–1494, 1992.

[28] W. H. Merigan and L. M. Katz. Spatial resolution across the macaque retina. *Vision research*, 30(7):985–991, 1990.

[29] M. Mylo, M. Giesel, Q. Zaidi, M. Hullin, and R. Klein. Appearance bending: A perceptual editing paradigm for data-driven material models. In *Vision, Modeling & Visualization*, 2017.

[30] P. Nair, A. Popli, and K. N. Chaudhury. A fast approximation of the bilateral filter using the discrete fourier transform. *Image Processing On Line*, 7:115–130, 2017.

[31] M. Narwaria, R. K. Mantiuk, M. P. Da Silva, and P. Le Callet. Hdr-vdp-2.2: a calibrated method for objective quality prediction of high-dynamic range and standard images. *Journal of Electronic Imaging*, 24(1):010501–010501, 2015.

[32] I. Newton. Opticks: Or a treatise of the reflexions, refractions, inflexions and colours of light. 1704.

[33] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607, 1996.

[34] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997.

[35] S. Paris, S. W. Hasinoff, and J. Kautz. Local laplacian filters: Edge-aware image processing with a laplacian pyramid. *ACM Trans. Graph.*, 30(4):68–1, 2011.

[36] S. Paris, P. Kornprobst, J. Tumblin, F. Durand, et al. Bilateral filtering: Theory and applications. *Foundations and Trends® in Computer Graphics and Vision*, 4(1):1–73, 2009.

[37] R. Rajalingham, K. Schmidt, and J. J. DiCarlo. Comparison of object recognition behavior in human and monkey. *Journal of Neuroscience*, 35(35):12127–12136, 2015.

[38] D. L. Ringach. Spatial structure and symmetry of simple-cell receptive fields in macaque primary visual cortex. *Journal of neurophysiology*, 88(1):455–463, 2002.

[39] D. L. Ringach, G. Sapiro, and R. Shapley. A subspace reverse-correlation technique for the study of visual neurons. *Vision research*, 37(17):2455–2464, 1997.

[40] E. Salinas and L. Abbott. Do simple cells in primary visual cortex form a tight frame? *Neural computation*, 12(2):313–335, 2000.

[41] T. D. Sanger. Analysis of the two-dimensional receptive fields learned by the generalized hebbian algorithm in response to random input. *Biological cybernetics*, 63(3):221–228, 1990.

[42] B. Selby. Development of an integrated model of primary visual cortex. Master's thesis, University of Waterloo, 2016.

[43] R. N. Shepard. Perceptual-cognitive universals as reflections of the world. *Behavioral and brain sciences*, 24(4):581–601, 2001.

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

[45] D. Turcsány. *Deep learning models of biological visual information processing*. PhD thesis, University of Nottingham, 2016.

[46] A. B. Watson et al. The cortex transform- rapid computation of simulated neural images. *Computer vision, graphics, and image processing*, 39(3):311–327, 1987.

[47] M. Weinmann, J. Gall, and R. Klein. Material classification based on training data synthesized using a btf database. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part III*, pages 156–171. Springer International Publishing, 2014.

Multiscale Fully Convolutional DenseNet for Semantic Segmentation

Sourour BRAHIMI
REGIM-Lab: Research Groups in Intelligent Machines, University of Sfax, National School of Engineers of Sfax (ENIS), BP 1173, Sfax, 3038, Tunisia
sourour.brahimi.TN@ieee.org

Najib BEN AOUN
REGIM-Lab: Research Groups in Intelligent Machines, University of Sfax, National School of Engineers of Sfax (ENIS), BP 1173, Sfax, 3038, Tunisia
Department of Computer Science, College of Computer Science and Information Technology, AL-BAHA University, Saudi Arabia
najib.benaoun@ieee.org

Chokri BEN AMAR
REGIM-Lab: Research Groups in Intelligent Machines, University of Sfax, National School of Engineers of Sfax (ENIS), BP 1173, Sfax, 3038, Tunisia
chokri.benamar@ieee.org

Alexandre BENOIT, Patrick LAMBERT
LISTIC-Lab: Univ. Savoie Mont Blanc, LISTIC, Polytech Annecy Chambéry, 5 ch. de Bellevue, Annecy-le-Vieux, 74940, Annecy, France
{alexandre.benoit, patrick.lambert}@univ-smb.fr

ABSTRACT

In the computer vision field, semantic segmentation represents a very interesting task. Convolutional Neural Network methods have shown their great performances in comparison with other semantic segmentation methods. In this paper, we propose a multiscale fully convolutional DenseNet approach for semantic segmentation. Our approach is based on the successful fully convolutional DenseNet method. It is reinforced by integrating a multiscale kernel prediction after the last dense block which performs model averaging over different spatial scales and provides more flexibility of our network to presume more information. Experiments on two semantic segmentation benchmarks: CamVid and Cityscapes have shown the effectiveness of our approach which has outperformed many recent works.

Keywords

Semantic Segmentation, Convolutional Neural Network, Fully Convolutional DenseNet, Dense Block, MultiScale Kernel Prediction.

1 INTRODUCTION

Today, semantic segmentation represents is very active topic in the computer vision field. It aims to group image pixels into semantically meaningful regions. It has been used for many applications such as video action and event recognition [Wal10a, Ben11a, Ben14a, Ben14b, Mej15a], image search engines [Wan14a, Ben10a], augmented reality [Alh17a], image and video coding [Ben11b, Ben12a], facial expression recognition [Bou16a], image retrieval [Sim14a] and autonomous robot navigation [Lin17a]. In recent years, a big gains in semantic segmentation have been obtained through the use of deep learning. In particular, the Convolutional Neural Network (CNN) methods [Lon15a, Bad15a, Jég17a, Wu16a] have given good semantic segmentation results due to their high capacity for data learning. As a result, many CNN variants have been developed such as Fully Convolutional Network (FCN) [Lon15a], deep fully convolutional neural network architecture for semantic pixel-wise segmentation (SegNet) [Bad15a], Wide Residual Network [Wu16a] and Fully convolutional DenseNet (FC-DenseNet) [Jég17a]. Specifically, FC-DenseNet method has substantially outperformed the prior state of the art methods on many datasets of the semantic segmentation task. Today, semantic segmentation

https://doi.org/10.24132/JWSCG.2018.26.2.5

represents is very active topic in the computer vision field. It aims to group image pixels into semantically meaningful regions. It has been used for many applications such as video action and event recognition [Wal10a, Ben11a, Ben14a, Ben14b, Mej15a], image search engines [Wan14a, Ben10a], augmented reality [Alh17a], image and video coding [Ben11b, Ben12a], facial expression recognition [Bou16a], image retrieval [Sim14a] and autonomous robot navigation [Lin17a]. In recent years, a big gains in semantic segmentation have been obtained through the use of deep learning. In particular, the Convolutional Neural Network (CNN) methods [Lon15a, Bad15a, Jég17a, Wu16a] have given good semantic segmentation results due to their high capacity for data learning. As a result, many CNN variants have been developed such as Fully Convolutional Network (FCN) [Lon15a], deep fully convolutional neural network architecture for semantic pixel-wise segmentation (SegNet) [Bad15a], Wide Residual Network [Wu16a] and Fully convolutional DenseNet (FC-DenseNet) [Jég17a]. Specifically, FC-DenseNet method has substantially outperformed the prior state of the art methods on many datasets of the semantic segmentation task.

In this paper, we propose a Multiscale FC-DenseNet (MS-DenseNet) which exploits the success of FC-DenseNet [Jég17a] for the semantic segmentation. Our method is built upon the FC-DenseNet and it is reinforced by integrating a MultiScale Kernel Convolutional (MSConv) layer after the last Dense Block (DB). The idea behind the use of multiscale kernel is inspired from [Aud16a]. Indeed, this layer aggregates information from 3 parallel convolutions with different kernel sizes in order to collect different spatial contexts. Moreover, it ensures more flexibility of our network to presume more information. Our MS-DenseNet was tested on two challenging benchmarks for semantic segmentation: CamVid [Bro09a] and Cityscapes [Cor15a] datasets. It has significantly improved the segmentation accuracy compared to all reported methods for both datasets.

The rest of our paper is organized as follows. The related works are reviewed in section 2. Then, in section 3, our proposed MS-DenseNet for semantic segmentation will be described. In section 4, the experimental results are presented for the two semantic segmentation benchmarks. Finally, in section 5, conclusions and some future directions are given.

2 RELATED WORKS

Due to the importance of the semantic segmentation, different methods have been developed such as: Graph based methods [Pou15a, Zha14a], Sparse Coding based methods [Zou12a] and CNN based methods [Lon15a, Bad15a, Jég17a, Wu16a]. In this section, we will focus our study on the CNN based methods [Lon15a, Bad15a, Jég17a, Wu16a, Bra16a] since they have shown their good performance and given the best segmentation and recognition results in recent works. The powerful of each CNN variant depends on the network architecture which makes two categories. The first category concerns the CNN methods that have been developed for classification task and extended to the semantic segmentation. The second category groups the encoder-decoder based CNN methods. These CNN methods are composed of two main parts. The first encoder part is similar to the architecture of the conventional CNN methods without neither the fully connected layers nor the classification layer. While the second decoder part is added in order to map the low resolution feature maps of the encoder to complete the input resolution feature maps. This is conducted for pixel-wise classification.

For the first category, image segmentation is conducted using adapted version of the classification oriented CNN methods. Long et al. [Lon15a] have proposed a Fully Convolutional Networks (FCN) method. This method consists of replacing the fully connected layers by convolutional layers with very large receptive fields. This will allow to detect and extract the global context of the scene and output spatial heat maps. It has been built upon AlexNet [Kri12a], VGG-16 [Sim14a] and GoogLeNet [Sze15a]. Figure. 1 presents the FCN-AlexNet architecture. In addition, a ReSeg [Vis16a] method was proposed. This method has extended the ReNet [Vis15a] classification method to the semantic segmentation. It is composed of four Recurrent Neural Networks (RNNs) which retrieve the contextual information by scanning the image in both horizontal and vertical directions. Then, the last feature map is re-sized by one or more max-pooling layers. Finally, to presume the probability distribution over the classes for each pixel, a soft-max layer is used.

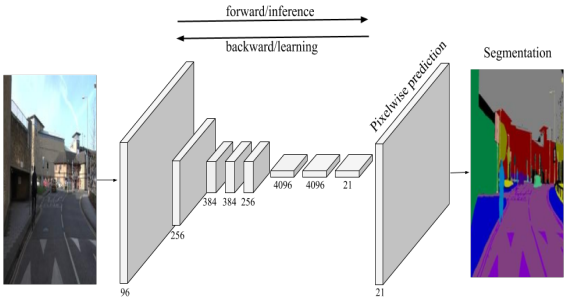


Figure 1: Fully Convolutional Networks (FCN) architecture

Despite their success, the extensions of the conventional CNN methods did not succeed to overcome the problem of learning to decode low-resolution images to pixel-wise predictions for segmentation. That is why, an encoder-decoder architecture was proposed. SegNet [Bad15a] is an example of encoder-decoder methods

(see Figure. 2). It is composed of two symmetric parts where the decoder is an exact mirror of the encoder. The encoder part is composed of 13 convolutional layers inspired from VGG-16 [Sim14a] method. Then, the encoder has a corresponding decoder part with 13 layers which maps the low resolution feature maps of the encoder. Finally, a soft-max classifier is used in order to produce class probabilities for each pixel independently.

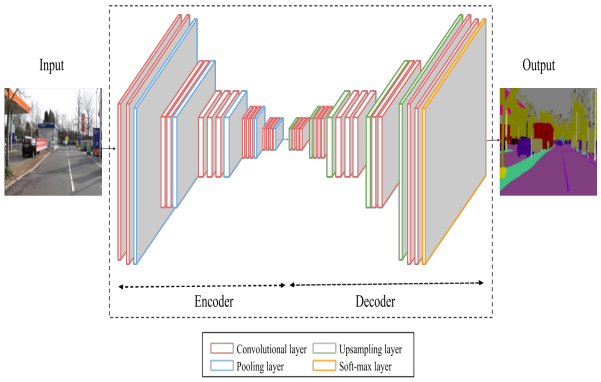


Figure 2: Example of SegNet architecture

Besides, the Efficient Neural Network (ENet) [Pas16a] has been introduced as an encoder-decoder CNN method which has a large encoder and small decoder parts. Each block in ENet architecture is composed of three convolutional layers. Batch Normalization (BN) as well as the Parametric Rectified Linear Unit (PReLU) has been placed between all convolutions. In addition, DeepLab [Che14a] applied an atrous convolutional with up-sampled filters for dense feature extraction. This method exploits deep CNN and fully connected conditional random fields in order to improve the localization performance. Their main idea is to incorporate larger context by enlarging the view field. Moreover, a Dilated convolution method is proposed in [Yu15a] with a dilated filter. This dilated filter is adapted to dense prediction without losing the resolution. It is composed of dilated convolutional layers which aggregate a multiscale contextual information.

Recently, Simon J. et al. have proposed an FC-DenseNet [Jég17a] method which transformed the existing classification model DenseNet [Gao16a] into fully convolutional one. FC-DenseNet is composed of 11 dense blocks (DBs) with five DBs in the encoder part, one DB in the BottleNeck (between the encoder and the decoder) and 5 DBs in the decoder part. In fact, each DB is composed of BN, Rectified Linear Unit (ReLU) layer and a 3×3 convolutional layer. Besides, the DB integrates direct connections from any layer to all subsequent layers. In the encoder part, each DB is followed by a Transition Down (TD) transformation which is composed of BN, ReLU, a 1×1 convolutional layer and a 2×2 max pooling operation. The layer between the encoder and the decoder is referred to as bottleneck. However, in the decoder part each DB is followed by a Transition Up (TU) transformation which is composed of a 3×3 transposed convolution and a stride equal to 2. The transposed convolution consists on upsampling the previous feature maps. Then, the feature maps outputted from the TU layer are concatenated together with the feature maps received from the skip connection. The result of this concatenation will form the input for a new dense block. Finally, a 1×1 convolutional layer followed by Softmax classification method are used to give the per class distribution at each pixel. Figure. 3 visualizes the architecture of FC-DenseNet with only 5 DB, 2 TD and 2 TU.

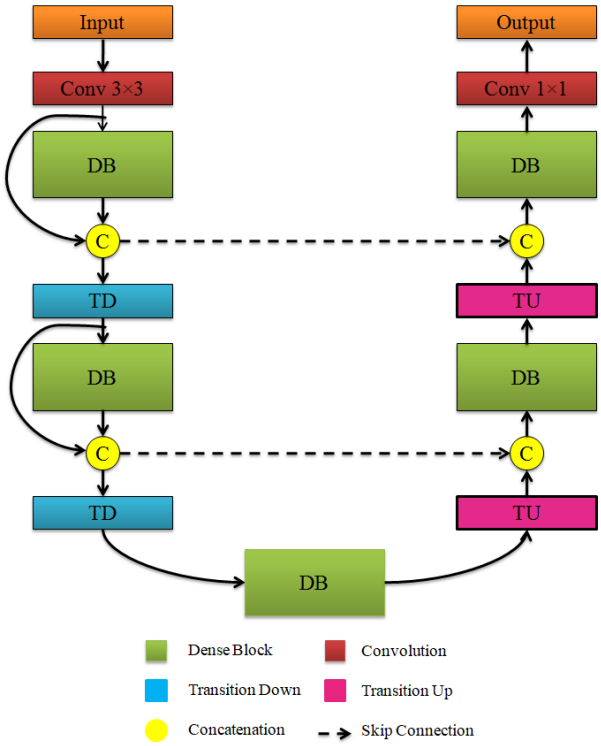


Figure 3: Fully convolutional DenseNet architecture with only five dense blocks. "c" stands for concatenation and interrupted lines are skip connections.

Among the reported methods, FC-DenseNet [Jég17a] has experimentally proven its power for many image segmentation benchmarks. That is what encourages us to build our proposed method on the FC-DenseNet.

3 PROPOSED APPROACH

The central idea of our MultiScale fully convolutional DenseNet (MS-DenseNet) is to take advantage of the FC-DenseNet [Jég17a] method while using multiscale Kernel prediction for the semantic segmentation task. Indeed, a MSConv layer is added to ensure more flex-

ibility of our network and to presume more information. It is conducted to boost the performance of our network. Table 1 details the architecture of our MS-DenseNet method.

3.1 MultiScale Fully Convolutional DenseNet Architecture

As it can be seen in Table 1, our MS-DenseNet is build from 96 convolutional layers: one convolutional layer in the input, 38 layers in the encoder part, 15 layers in the bottleneck, 38 layers in the decoder part with one MSConv layer and one convolutional layer at the end (See Table 1). First, the input image is passed through a standard convolutional layer with 3 ×3 receptive field. Then, 5 DBs are conducted in the encoder part, one DB in the BottleNeck and 5 DBs in the decoder part. As shown in (see Figure. 4), each DB is composed of BN, Rectified Linear Unit (ReLU) layer and a 3×3 convolutional layer. The DB integrates direct connections from any layer to all succeding layers. In the encoder part each DB is followed by a Transition Down (TD) transformation (see Table 1). Each TD is composed of BN, ReLU, a 1×1 convolutional layer, dropout (with p = 0.2) and a 2×2 max pooling operation (see Figure. 4). In the decoder part each DB is followed by a Transition Up (TU) transformation. Each TU is composed of a 3×3 transposed convolution (stride=2) in order to compensate the pooling operation (see Figure. 4). In order to perform model averaging over several scales, MSConv layer is conducted after the last DB. Finally, a convolutional layer with 1 ×1 receptive field and a Soft-max layer are used to determine the inclusion of each pixel to each class.

3.2 MultiScale Kernel Convolutional Layer

Following the multiscale convolutional architecture used in [Aud16a], we have applied a MSConv layer (see Figure. 5) in our method. This layer performs 3 parallel convolutions using different kernels with 1×1 , 3×3 and 5×5 receptive fields contrarily to FC-DenseNet [Jég17a] method that uses only one kernel with 1×1 size. As a result, three different feature maps will be obtained. They will be concatenated together into one feature map. By a conducting these three parallel convolutional layers, our model will aggregate the predictions at different scales while giving only one prediction output. Using MSConv layer, our network becomes more flexibile to presume more information and it will improve the segmentation accuracy.

4 EXPERIMENTAL RESULTS

In this section, we will provide the experimental details. Our proposed method was initialized using HeUniform [He15a] and trained with RMSprop [Tie12a], with an initial learning rate of 0.001. Our approach is evaluated on two datasets used as benchmarks for semantic segmentation: CamVid [Bro09a] and Cityscapes [Cor15a]. For these two datasets, the Mean Intersection over Union (mIoU) is used as a metric to measure the segmentation performance. The IoU determines the similarity between the ground-truth region and the predicted region for an object present in the image. The mean IoU (mIoU) is simply the average over all classes. The IoU is defined to a given class c, predictions (p_i) and targets (t_i), by:

$$IoU(c) = \frac{\sum_i (p_i = c \wedge t_i == c)}{\sum_i (p_i == c \vee t_i == c)} \quad (1)$$

Layer
Batch Normalization
ReLU
3×3 Convolution
Dropout p = 0.2

Transition Down (TD)
Batch Normalization
ReLU
1×1 Convolution
Dropout p = 0.2
2×2 Max Pooling

Transition Up (TU)
3×3 Transposed Convolution stride = 2

Figure 4: Different blocks of MS-DenseNet: the layer used in the model, the Transition Down (TD) and the Transition Up (TU).

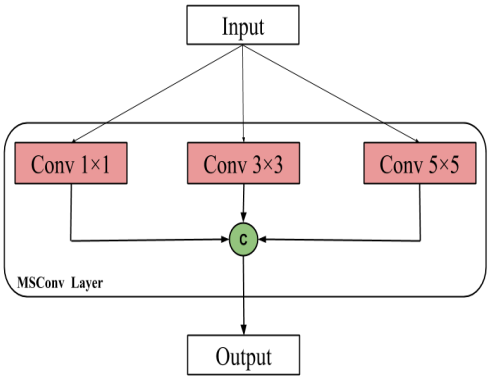


Figure 5: MultiScale Kernel Convolutional Layer

	MS-DenseNet	
	Layers	Configuration
Encoder	Convolution	3×3 Conv
	DB	4 layers
	Transition Down	
	DB	5 layers
	Transition Down	
	DB	7 layers
	Transition Down	
	DB	10 layers
	Transition Down	
	DB	12 layers
	Transition Down	
Bottleneck	DB	15 layers
Decoder	Transition Up	
	DB	12 layers
	Transition Up	
	DB	10 layers
	Transition Up	
	DB	7 layers
	Transition Up	
	DB	5 layers
	Transition Up	
	DB	4 layers
	MSConv	MultiScale Convolution
	Convolution	1×1 Conv
	Segmentation layer	Softmax

Table 1: MS-DenseNet Architecture

where $\wedge$ represents the logical "and" operation, and $\vee$ represents the logical "or" operation. The IoU is computed by summing over all the pixels i of the dataset. Besides, our MS-DenseNet method was implemented using the publicly available TensorFlow Python API [Aba16a].

4.1 CamVid dataset

Cambridge-driving Labeled Video Database (CamVid) [Bro09a] is one of the most commonly used semantic segmentation dataset with 32 semantic classes. In fact, only 11 classes have been used for our experiments: sky, building, pole, road, sidewalk, vegetation, sign, fence, car, pedestrian, cyclist and void, in order to compare our system to recent methods [Lon15a, Pas16a, Bad15a, Vis16a, Ken15a, Jég17a, Yu15a]. This dataset contains 701 semantic segmentation frames: 367 frames used to train the network, 233 for testing and 101 for validation. The size of each frame is 360×480. Figure. 6 visualizes samples from CamVid dataset. Our MS-DenseNet method was trained with image crops of 224×224. The maximum mIoU score has been reached with 225985 steps with 256 batch size.

Table 2 presents the mIoU scores of our method in comparison with the recent semantic segmentation methods in the literature. ENet [Pas16a] has given a lower result than other methods. In addition, FCN-8 [Lon15a] has also failed to give acceptable segmentation results. This can be explained by the fact that the spatial invariance does not take into account useful context execution information. Moreover, Reseg [Vis16a] which takes the advantages of RNN, gives low results less than 59%. Similarly, SegNet [Bad15a] which is an encoder-decoder based model, has given weak results because of the inefficient CNN configuration used. However, despite the improvement done by using Bayesian filters within the Bayesian SegNet [Ken15a] method, the result is still limited. This

Figure 6: Samples from CamVid dataset

network suffers from the speed degradation problem. Besides, Dilation [Yu15a], which has incorporated long spatio-temporal regularization to the output of FCN-8 to boost their performance, has given promising result with 65.30% mIoU scores. Among the state of the art methods, FC-DenseNet [Jég17a] has given the highest mIoU score (66.90%). It is based essentially on DenseNet [Gao16a] classification method. That is why our MS-DenseNet method followed the same architecture while integrating MSConv layer. Adding MSConv layer has given a very promising result. It gives an mIoU score gain of 1.21% compared to the FC-DenseNet method and reaches 68.11%. It proves more that our MS-Densenet architecture is very promising. Examples of images segmented using our MS-DenseNet method are shown in Figure. 7

Model	mIoU (%)
ENet [Pas16a]	55.60
FCN-8 [Lon15a]	57.00
ReSeg[Vis16a]	58.80
SegNet [Bad15a]	60.10
Bayesian SegNet [Ken15a]	63.10
Dilation [Yu15a]	65.30
FC-DenseNet [Jég17a]	66.90
MS-denseNet	**68.11**

Table 2: Results on CamVid evaluation set

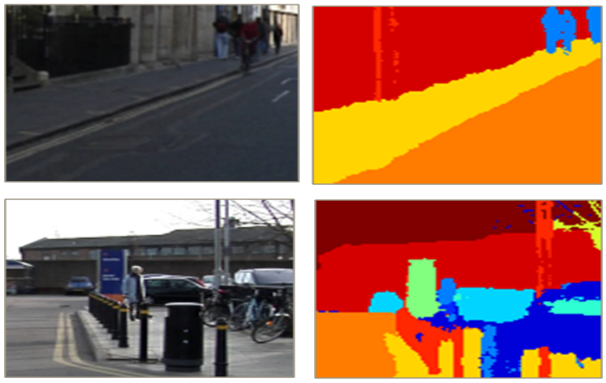


Figure 7: Qualitative results on the CamVid dataset

4.2 Cityscapes dataset

Cityscapes dataset [Cor15a] consists of 5000 images split into three sets: 2975 images for trainings, 500 for validation and 1525 for testing. It has a high image resolution 2048×1024 with 19 classes. Figure. 8 visualizes samples from Cityscapes dataset. The optimal result has been reached when the number of steps was 431425 with 256 batch size.

Table 3 presents a comparison between our method and the other reported methods performances on

Figure 8: Samples from Cityscapes dataset

CityScapes. Similarly to the CamVid dataset, ENet [Pas16a] and FCN-8 [Lon15a] have given weak results. Moreover, Dilation [Yu15a] method has given a 67.10 % mIoU score. Furthermore, different ResNet [He16a] based models such as DeepLab [Che14a], wide-ResNet [Wu16a] have given 70.40% and 78.40% respectively. Indeed, our MS-Densenet method has overcome all state of the art methods by a margin of 0.8% compared to the best reported one and gives 79.20%. This result confirms one more time the strength of our method.

Model	mIoU (%)
ENet [Pas16a]	58.30
FCN-8 [Gar17a]	65.30
Dilation [Yu15a]	67.10
DeepLab [Che14a]	70.40
Wide-ResNet [Wu16a]	78.40
MS-denseNet	**79.20**

Table 3: Results on CityScapes dataset

5 CONCLUSION AND FUTURE WORK

In this paper, a MultiScale FC-DenseNet method is proposed. It is built upon the FC-DenseNet while adding MultiScale kernel Convolutional layer. In fact, a MultiScale Kernel Convolutional layer is integrated after the last dense block in order to give a rich contextual prediction as well as to improve the results. Our method has been experimentally validated on two semantic segmentation benchmarks and has shown very promising results. Our plan for the future work is to improve our MS-DenseNet by optimizing its architecture.

6 ACKNOWLEDGEMENTS

The research leading to these results has received funding from the Ministry of Higher Education and Scientific Research of Tunisia under the grant agreement number LR11ES48. LISTIC experiments have been

made possible thanks to the MUST computing center of the University of Savoie Mont Blanc.

7 REFERENCES

[Wal10a] Wali, A., Ben Aoun, N., Karray, H., Ben Amar, C., and Alimi, A. M., A New System for Event Detection from Video Surveillance Sequences. In ACIVS, pp. 110-120, 2010.

[Wan14a] Wan, J, Wang, D, Hoi, S.C.H., Wu, P, Zhu, J, Zhang, Y and Li, J, Deep learning for content-based image retrieval: A comprehensive study. In ACM international conference on Multimedia, pp.157-166, 2014.

[Ben12a] Ben Aoun, N., Elarbi, M., and Ben Amar, C., Wavelet Transform Based Motion Estimation and Compensation for Video Coding. Advances in Wavelet Theory and Their Applications in Engineering, Physics and Technology, Dr. Dumitru Baleanu (Ed.), pp. 23-40, 2012.

[Ben14a] Ben Aoun, N., Mejdoub, M., Ben Amar, C., Graph-based approach for human action recognition using spatio-temporal features. Journal of Visual Communication and Image Representation, 25 (2): 329-338, 2014.

[Alh17a] Alhaija, H.A., Mustikovela, S.K., Mescheder, L., Geiger, A., and Rother, C., Augmented Reality Meets Deep Learning for Car Instance Segmentation in Urban Scenes. In British Machine Vision Conference, Vol. 3,2017.

[Ben11a] Ben Aoun, N., Elghazel, H., and Ben Amar, C., Graph modeling based video event detection. In IIT, pp. 114-117, 2011.

[Lin17a] Lin, J., Wang, W.J., Huang, S.K., and Chen, H.C., Learning based semantic segmentation for robot navigation in outdoor environment. In IFSA-SCIS, pp. 1-5, 2017.

[Pou15a] Pourian, N., Karthikeyan, S., and Manjunath, B.S., Weakly supervised graph based semantic segmentation by learning communities of image-parts. In Proceedings of the ICCV, pp. 1359-1367, 2015.

[Mej15a] Mejdoub, M., Ben Aoun, N. and Ben Amar, C., Bag of frequent subgraphs approach for image classification. Intelligent Data Analysis 19 (1): 75-88, 2015.

[Zha14a] Zhang, K., Zhang, W., Zeng, S., and Xue, X., Semantic Segmentation Using Multiple Graphs with Block-Diagonal Constraints. In AAAI, pp. 2867-2873, 2014.

[Ben14b] Ben Aoun, N., Mejdoub, M., and Ben Amar, C., graph-based video event recognition. In ICASSP, pp. 1566-1570, 2014.

[Zou12a] Zou, W., Kpalma, K., and Ronsin, J. Semantic segmentation via sparse coding over hierarchical regions. In ICIP, pp. 2577-2580, 2012.

[Lon15a] Long, J., Shelhamer, E., and Darrell, T., Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE CVPR, pp. 3431-3440, 2015.

[Bra16a] Brahimi, S., Ben Aoun, N., Ben Amar, C., Very deep recurrent convolutional neural network for object recognition. In ICMV, 2016.

[Ben10a] Ben Aoun, N., Elarbi, M., and Ben Amar C., Multiresolution motion estimation and compensation for video coding. In ICSP, pp. 1121-1124, 2010.

[Kri12a] Krizhevsky, A., Sutskever, I., and Hinton, G., Imagenet classification with deep convolutional neural networks. In NIPS, pp. 1097-1105, 2012.

[Sze15a] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A., Going deeper with convolutions. In CVPR, pp.1-9, 2015.

[Bad15a] Badrinarayanan, V., Kendall, A., and Cipolla, R., Segnet: A deep convolutional encoder-decoder architecture for image segmentation, 2015. arXiv preprint arXiv:1511.00561, 2015.

[Ben11b] Ben Aoun, N., Elghazel, H., Hacid, M.S., and Ben Amar, C., Graph aggregation based image modeling and indexing for video annotation. In CAIP, pp. 324-331, 2011.

[Jég17a] Jégou, S., Drozdzal, M., Vazquez, D., Romero, A., and Bengio, Y., The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In CVPRW, pp. 117-1183, 2017, July.

[Aud16a] Audebert, N., Le, Saux, B., and Lefevre, S., Semantic segmentation of earth observation data using multimodal and multi-scale deep networks. In Asian Conference on Computer Vision, pp. 180-196, 2016.

[Wu16a] Wu, Z., Shen, C., and Hengel, A.V.D., Wider or deeper: Revisiting the resnet model for visual recognition, 2016. arXiv preprint arXiv:1611.10080.

[Bro09a] Brostow, G.J., Fauqueur, J., and Cipolla, R., Semantic object classes in video: A high-definition ground truth database. Pattern Recognition Letters, 30(2): 88-97, 2009.

[Cor15a] Cordts, M., Omran, M., Ramos, S., Scharwachter, T., Enzweiler, T., Benenson, R., Franke, U., Roth, S., and Schiele, B., The cityscapes dataset. In CVPR, 2015.

[Sim14a] Guedri, B., Zaied, M., Ben Amar, C., Indexing and images retrieval by content. In : High Performance Computing and Simulation (HPCS), 369-375, 2011.

[Sim14a] Simonyan, K., and Zisserman, A., Very deep convolutional networks for large-scale image recog-

nition, 2014. arXiv preprint arXiv:1409.1556.

[Vis16a] Visin, F., Ciccone, M., Romero, A., Kastner, K., Cho, K., Bengio, Y., Matteucci, M., and Courville, A., Reseg: A recurrent neural network-based model for semantic segmentation. In CVPR, pp. 426-433, 2016.

[Vis15a] Visin, F., Kastner, K., Cho, K., Matteucci, M., Courville, A.C., and Bengio, Y., Renet: A recurrent neural network based alternative to convolutional networks, 2015. arXiv:1505.00393v3.

[Pas16a] Paszke, A., Chaurasia, A., Kim, S., and Culurciello, E., Enet: A deep neural network architecture for real-time semantic segmentation, 2016. arXiv preprint arXiv:1606.02147.

[Che14a] Chen, L.C., Papandreou, G, Kokkinos, I., Murphy, K., and Yuille, A.L., DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs, 2014. arXiv preprint arXiv:1606.00915.

[Yu15a] Yu, F., and Koltun, V., Multi-scale context aggregation by dilated convolutions, 2015. arXiv preprint arXiv:1511.07122.

[Gao16a] Gao, H., Zhuang, L., and Kilian, Q.W., Densely connected convolutional networks, 2016. arXiv:1608.06993v3.

[He16a] He, K., Zhang, X., Ren, S., and Sun, J., Deep residual learning for image recognition. In CVPR, pp. 770-778, 2016.

[He15a] He, K., Zhang, X., Ren, S., and Sun, J., Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE international conference on computer vision, pp. 1026-1034, 2015.

[Tie12a] Tieleman, T., and Hinton, G., rmsprop adaptive learning. In COURSERA: Neural Networks for Machine Learning, 2012.

[Aba16a] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., and Ghemawat, S., Tensorflow: Large-scale machine learning on heterogeneous distributed systems. Publicly available at: https://tensorflow.org

[Gar17a] Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., and Garcia-Rodriguez, J., A Review on Deep Learning Techniques Applied to Semantic Segmentation, 2017. arXiv preprint arXiv:1704.06857.

[Ken15a] Kendall, A., Badrinarayanan, V., and Cipolla, R., Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding, 2015. arXiv:1511.02680.

[Bou16a] Boughrara, H., Chtourou, M., Ben Amar, C. , Chen, L., Facial expression recognition based on a mlp neural network using constructive training algorithm, Multimedia Tools and Applications, 75(2), 709-731, 2016.

Morphological Amoeba-based Patches for Exemplar-Based Inpainting

Susana Castillo[1] Douglas W. Cunningham[1] Christian Winger[2] Michael Breuß[3]

[1] BTU Cottbus-Senftenberg
Konrad-Wachsmann-Allee 5,
03046 Cottbus, Germany
castillo@b-tu.de
douglas.cunningham@b-tu.de

[2] Öko-Institut e.V., Germany
Merzhauser Straße 173
79100 Freiburg, Germany
c.winger@oeko.de

[3] BTU Cottbus-Senftenberg
Platz der Deutschen Einheit 1,
03046 Cottbus, Germany
breuss@b-tu.de

ABSTRACT

Inpainting is the process of replacing areas in an image with a perceptually plausible substitution. A common technique is to iteratively match and fill small patches at the edge of the target region making use of similar patches from the same image. Nearly all inpainting algorithms based on this approach use a single patch size for the entire image. Yet, it seems clear that differently sized structures within the same image – for example a leaf versus a car tire – may require different patch sizes in order to achieve reasonable inpainting results. Likewise, a fixed patch size will give different results for the same image when the image resolution is doubled. A reasonable patch should therefore take into account the overall image size as well as the size and shape of the structures at the patch location. The aim of our paper is to study the effect of adaptively altering size and shape of the patch. We show that this technique leads to a better quality of the inpainting result compared to a fixed patch size.

Keywords

Inpainting, Adaptivity, Criminisi algorithm, Morphological Amoeba

1 INTRODUCTION

The class of techniques designed to replace empty regions in an image with perceptually plausible content is called inpainting after Bertalmió et al. [Ber00a]. Inpainting can be used for many purposes in visual computing, including, for example, denoising [Ad17a], image compression [Mai09], or automatic repair of damaged images [Cai17a]. There are many technical approaches to inpainting, cf. [Gui14a] for an overview. These techniques include exploring information from level lines [Mas98a], tackling the task as a texture synthesis problem [Ef99a], or making use of partial differential equations (PDEs) [Ber00a]. One may also combine different techniques, usually with improved results [Ber03a]. Given the wide variety of potential applications and approaches, it is of fundamental interest to explore and understand the different building blocks of the most promising inpainting methods.

One of the most central works in image inpainting is Criminisi et al. [Cri04a] whose algorithm has since become the core of most exemplar-based approaches (see, e.g., [Buy15a] for a broader discussion of the approach). Exemplar-based approaches assume that the best description of the information to be filled in can be found somewhere else in the same image. Exemplar-based inpainting methods follow a general pipeline. First, the border of the empty or target region is located. Second, a pixel on the border is selected and a small patch is centered on the selected pixel. Note that part of the patch will contain valid image information and part will be in the target region. The size of the patch is set manually, with the size usually chosen to match the largest relevant feature in the image. Traditionally, patch sizes of 5×5, 7×7, 9×9 and sometimes 11×11 pixels are used [Lem13a]. The third step, filling-in, is subdivided into finding a matching patch and copying the new patch into the target patch. This process is iterated until all holes have been filled. A number of newer algorithms have altered individual building blocks in Criminisi et al.'s [Cri04a] pipeline. Modifications encompass attempts to produce better descriptions of the contents of a patch (e.g., [Lem11a, Xu10a]), constructing a more efficient matching process (e.g., [Xi13a, Ngu13a]) or proposing more elaborate texture propagation/copy procedures (e.g., [Kom07a, Lem13a]). Nearly all of the proposed

https://doi.org/10.24132/JWSCG.2018.26.2.6

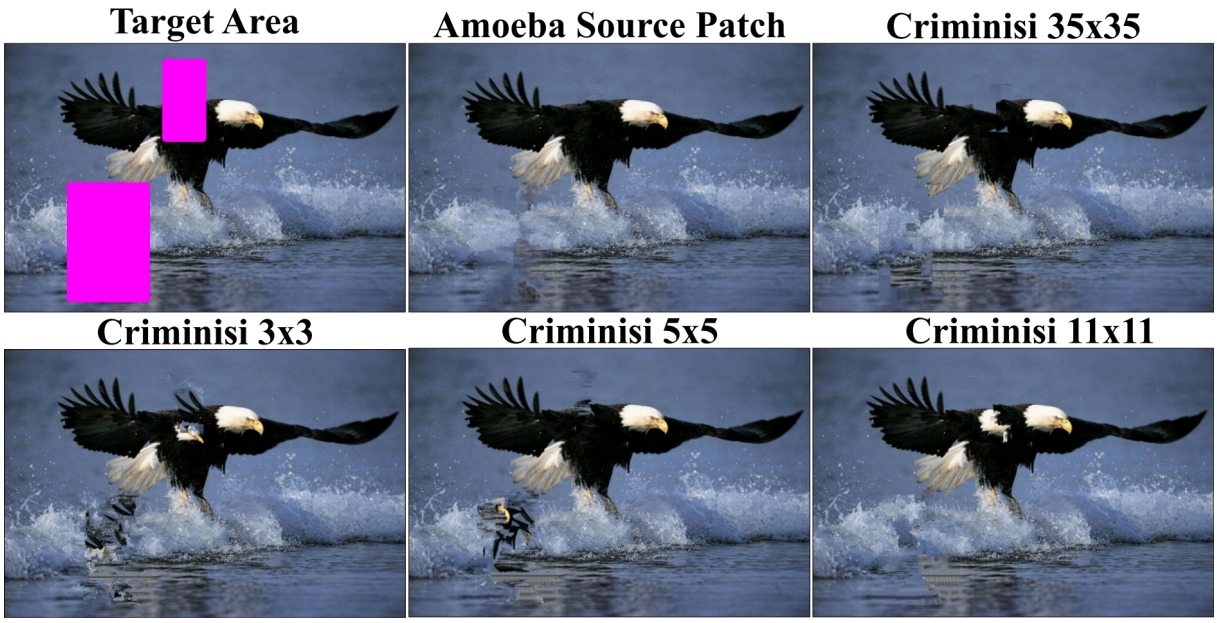


Figure 1: In reading order: Original image to be restored, the result of our new variable patch shape approach, and four results of an exemplar-based method – with two extreme (35×35 and 3×3) and two commonly used (5×5 and 11×11) patch-sizes

improvements leave the underlying patch concept unchanged, using a one-size-fits-all approach to patch size. While one can set the patch size to match image size or general trends in feature size, it appears evident that no uniform patch size and shape can capture the range of possible feature shapes. For example, structures such as long, bold edges, may require large, non-square patches. Structures that rapidly change such as a raged edge would require a smaller patch (see Figure 1).

The most similar work to the approach we will study is the work of Wu and Ruan, who proposed that patch size could be changed dynamically to match local texture information [Wu09a]. As usual, they created a small patch (4×4) around each pixel at the edge of the target region. They then calculated the color variation in each patch allowed to either grow (to 5×5) or shrink (to 3×3) based on a user-defined threshold. They argued that a lot of color variation probably represents a textured region and should therefore be assigned a larger patch in hopes of capturing more of the texture. A homogeneous region, on the other hand, will have no color variation. They claimed that patches for homogeneous regions should be kept small to prevent accidentally introducing structure. After filling-in, a divergence constrained PDE is used to reduce differences between neighboring patches.

In this paper, we propose that patch size should be based on the size and shape of local structures. Large structures should get large patches, small structures should get small patches. Likewise, non-square structures should get non-square patches. Since all previous work on exemplar-based inpainting exclusively used square patches, they generally either copied undesired structure (introducing artifacts) or copied partial structures (creating salient discontinuities). These *intrusion artifacts* can be seen in the fixed-patch size examples in Figure 1.

In **our contribution** we argue that, to completely capture the local structure at the boundary of the target region, the patch size should be iteratively altered until the local structure is fully enclosed. This would require that the method determines when the local structure is fully enclosed. Color variation as considered in [Wu09a] appears not to be able to capture the local image information, as the same variation in a patch's color may arise from a long edge or from a scattered texture. Thus, some form of feature extraction is needed. We follow the image segmentation ideas from [Ler07a] to create – at the source area – flexible, dynamic patches of arbitrary size and shape that capture and therefore copy only relevant structure. This focus on segmenting and copying the relevant structures helps to avoid intrusion artifacts and leads to visually pleasant results (see Figure 1).

2 ALGORITHM

Just like most exemplar-based inpainting techniques, ours is based on Criminisi et al.'s [Cri04a] algorithm, which showed that the order in which the target regions gets filled-in is important. Prioritizing patches in which structural elements are pointing inwards into the target area produces considerably better completions. Criminisi et al.'s algorithm is illustrated in Figure 2 and is

discussed in more detail in the remainder of this section.

In a pre-processing step, the image is converted into CIE L*a*b color space and all operations are performed on the three color channels simultaneously. In the first step, a target pixel located along the border of the target region $\partial\Omega$ is chosen based on the priority values. Then, a target patch $\Psi_{\hat{p}}$ (represented in Figure 2b by the dashed square at the border of the target region $\partial\Omega$) is created surrounding the target pixel (represented by a large black dot in Figure 2b). From all the possible source patches within the search space $q \in \Phi$, the best match $\Psi_{\hat{q}}$ is determined (see Figure 2c). Next, as shown in Figure 2d, the pixels in the target area portion of the target patch $\Psi_{\hat{p}} \cap \Omega$ are filled with the corresponding pixels from the source patch $\Psi_{\hat{q}}$. The border of the target region $\partial\Omega$ is then updated and the confidence values of the newly copied pixels are set. These steps are repeated until the target area Ω is filled.

Here, to more clearly study the effect of focusing on image features rather than image regions, we have chosen to stick as closely as possible to Criminisi et al.'s algorithm in all stages with the sole exception of the size and shape of the source patch $\Psi_{\hat{q}}$. Of course, as mentioned before, the modular nature of this pipeline means that our proposed change can be combined with any of the other modifications. Thus, we start with a fixed-size square patch surrounding the target pixel. After using this target patch to find the correct match, the center of the source patch is used as a seed for a region-growing segmentation technique called a *morphological amoeba*, which captures the structure at the source region (see Section 2.3). We then copy only the pixels within the newly-grown source amoeba to the corresponding locations in the target area Ω. In the next few subsections, we provide more detail about the individual steps of the full algorithm.

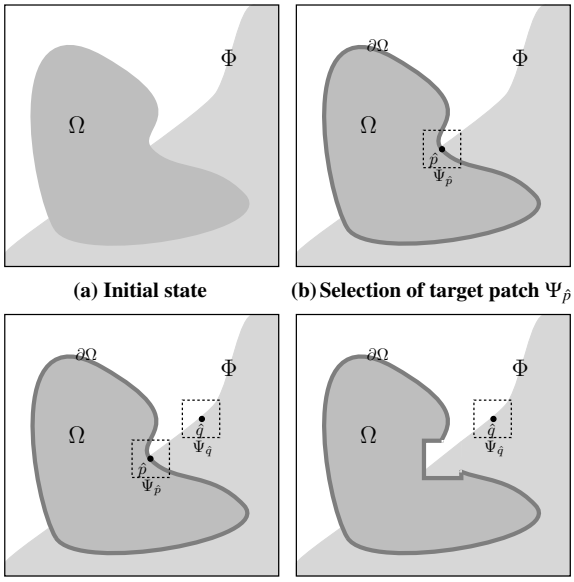
2.1 Target Patch Selection

Following Criminisi et al. [Cri04a], the first step in every iteration is the selection of the next target patch $\Psi_{\hat{p}}$ centering on the pixel with highest priority $\hat{p}$ at the contour $\partial\Omega$. All points $p \in \partial\Omega$ are sorted by a priority value $P(p)$, which takes into account two different factors, a data term and a confidence term:

$$P(p) = \varsigma(p) \cdot \gamma(p) \tag{1}$$

The confidence term $\gamma(p)$ is a measure of the reliability of the known image data around point p. It is the sum of the confidence value $C(i)$ of all the pixels i in the target patch divided by the number of pixels in the patch:

$$\gamma(p) = \frac{\sum_{i \in \Psi_p} C(i)}{\|\Psi_p\|} \tag{2}$$



(a) Initial state **(b) Selection of target patch $\Psi_{\hat{p}}$**

(c) Matching source patch $\Psi_{\hat{q}}$ **(d) Filling data in Ω**

Figure 2: Different steps of Criminisi's inpainting algorithm

Since all unknown pixels (the ones in the target area Ω) start with a confidence of 0 and all known pixels (the ones in the source area Φ) start with a confidence of 1, the more known pixels a patch has, the higher that patch's confidence value will be. After filling in, the newly filled-in pixels will receive a confidence less than 1 (see below for more details). As a result, patches close to the initial target-region border will have a higher confidence value than patches inside the (original) target area.

The data term $\varsigma(p)$ is a measure of the intensity of any linear structures pointing directly into the target area. A strong edge disappearing directly into the target area should be processed with higher priority than either a weak edge disappearing directly into the target area or a strong edge that is tangent to the target area. This will help to maintain the continuity of structural elements. The data term is defined as the normalized product of the dominant *isophote* $\nabla I^{\perp}$ in the target patch around a given point p and the normal vector $\vec{n}_p$ of the contour at that point:

$$\varsigma(p) = (\vec{n}_p \cdot \nabla I_p^{\perp})/\delta \tag{3}$$

with δ being a normalizing constant based on the possible pixel values and the isophote $\nabla I^{\perp}$ being defined as perpendicular to the intensity gradient. Thus, the direction of the isophote $\nabla I^{\perp}$ describes the orientation of the prevalent linear structure in the target area. To calculate the data term, we need to determine the dominant isophote in the target patch. To do this, we first calculate the isophotes for all pixels in the known portion of the target patch. Note that an isophote pointing directly upwards represents the same linear structure as

one pointing directly downwards, and as such we flip all isophotes with angles greater than 180 degrees so that they point in the opposite direction. We then examine the isophote histogram separately for each of the three color channels (with the angles now ranging between 0 and 179) and find the bin with the maximum summed gradient magnitude. These are the dominant isophotes of the patch, one for each color channel. The maximum of these three isophotes is chosen as the dominant isophote of the patch.

The pixel $p \in \partial\Omega$ with the highest priority value $P(p)$ is chosen as the center of the next target patch.

2.2 Source Patch Matching

To find the best match, we calculate the Euclidean color distance between each pixel in the known portion of the target patch $\Psi_{\hat{p}}$ and the corresponding pixels in all possible source patches. The source patch with the smallest summed color distance is chosen.

2.3 Amoeba

For this stage, which is novel to our algorithm, we start with the observation that copying a rectangular shaped region from the source area will have a non-zero probability of copying undesired structures (i.e, structures that are not the same as the one to which the target pixel belongs). Furthermore, once even a single pixel of an artifact has been copied to the target region, future in-painting iterations will consider the artifact to be a valid structure and will tend to complete the artifact. To avoid this, we propose that only the pixels in the source area that most closely resemble the target pixel itself (and thus are most likely to be on the same surface; see [Gi79a]) should be copied. We have chosen the morphological amoeba introduced in [Ler07a] to do this, as it utilizes both the physical distance between two pixels as well as the color distance. The amoeba determines which pixels belong to the patch by calculating the summed color distances and physical distances between pixels along a path. All pixels which can be connected to the target pixel by a path whose summed (color and spatial) distance is less than a given threshold are included in the patch. Conceptually, the amoeba starts by calculating the Euclidean color distance between all neighboring pixels p and q: $dist_{pixel}(p,q)$. It then calculates the summed color and physical distance $L(\sigma)$ between two pixels x and y along a given path σ:

$$L(\sigma) = \sum_{i=0}^{n-1} (PD + \lambda \cdot dist_{pixel}(x_i, x_{i+1})) \qquad (4)$$

where n is the number of pixels along the path and $\lambda \geq 0$ is a weighting factor which allows one to control the relative influence of the color distance over the physical distance. Note that the saliency of the physical distance between neighboring pixels in a Cartesian coordinate system is dependent on the viewing distance and the monitor resolution. Thus, the physical distance is the variable PD. Given the most common viewing distances and monitor sizes, the physical distance will typically be between 0.6 and 2. Typically, λ is set to 1, with the aim of ensuring that the physical distance and the color distance have equal saliency and equal importance. The final distance $d_\lambda(x,y)$ for two specific pixels is the path with the lowest $L(\sigma)$. Finally, all pixels y whose distance $d_\lambda(x,y)$ to the seed pixel x is below a user set threshold TH, belong to the amoeba.

2.4 Filling-in and Updating the Contour

In the next step, the data from the source patch is copied to the unknown portions of the target patch $\Psi_{\hat{p}} \cap \Omega$. Then, the confidence of the newly copied pixels is set to the average of all the pixels that are in an amoeba-shaped region centered on the target pixel. Finally, the target area Ω and the contour $\partial\Omega$ are updated and the priority values for all affected contour points are recalculated.

3 RESULTS

There are scenarios where image restoration methods will work best, such as images where the content to be generated lies in smooth or irregularly-textured areas, commonly present in natural images, or in areas that are not so likely to attract human attention. To gain more insight into effect of the new patch algorithm, we decided to use as a benchmark a set of considerably more challenging images. Specifically, we used part (see Figure 3) of the benchmark proposed by Rubinstein et al. [Rub10a]. This benchmark is known for containing images with attributes that represent a challenge for the objectives of preserving content and structure and preventing artifacts. Furthermore, we created the to-be-filled (target) areas in these images by selecting places that are most likely to be very challenging (e.g., that break local structures) and that are in areas most likely to attract human attention [Cas11a].

We submitted the 16 images to the original Criminisi et al. algorithm as well as to our modified version. We tested the effect of changing the size of the square target patch, using 20 different patch sizes. The twenty possible patch radii were in the range $r \in [1,20]$. Since a patch always included the target pixel and the number of pixels equal to the radius in each direction, this resulted in a range of patch sizes from 3×3 to 41×41 pixels. It is important to note that nearly all other tests of exemplar-based approaches use three patch sizes: 5×5, 7×7, 9×9 corresponding to the radii 2, 3 and 4. More rarely, 11×11 patches (radius of 5) are also used. We tested a *considerably larger range of patches* in order to more fully explore the range of image features that can be captured.

Figure 3: The 16 images used in our tests. The magenta-filled areas show the parts to be inpainted

For our algorithm, the maximum amoeba distance TH was set to 20 and the physical distance PD was set to 1.

The average run time of the amoeba-based algorithm is similar to the average run time of the unmodified original. It is important to note that the amoeba-based algorithm only needs to be run once to find a good patch size, whereas the Criminisi algorithm needs to be run once for each desired patch size.

4 VALIDATION

Ideally, in order to assess the quality of the results of both algorithms, a perceptual experiment with human participants should be conducted. Unfortunately, due to the large number of results to compared (with 20 patch sizes, 2 algorithms, and 16 images there are 640 resulting images), the duration of a perceptual experiment becomes untenable (e.g., a 2AFC preference task comparing each reconstruction for a given image to all the other reconstructions for the same image would require 12,480 trials per participant). Thus, we will use computational metrics to evaluate the image quality. Even if the metrics cannot replace the subjective evaluation of a human, they can be complementary and give us some insights for pre-filtering the results.

4.1 Metrics

The choice of metric is not a trivial issue. We can not use any metric which performs a pixel by pixel comparison since inpainting does not try to generate any specific texture, but instead focuses on perceptually plausible results. This means that a metric is needed that taps into the highly subjective issue of which image looks more plausible or natural. The choice of metric is further constrained by the central issue of patch size, as many existing image quality metrics break an image down into smaller patches and then analyze the image on a patch by patch basis. Unfortunately, these metrics all use a single, fixed patch size for any given image. It does not seem appropriate to use a fixed patch size to evaluate the effect of adaptive patch sizes. While it is appears interesting to construct a metric using adaptive patches, that is beyond the scope of this article.

Following [Rub10a, Cas11a], we selected two metrics that assess low-level differences between two images to give us an idea about the coherence of the image as a whole and how consistent the results are in comparison with the intact parts of the damaged image. These two metrics are Color Layout (CL) [Kat01a] and Edge Histogram (EH)[Man01a] (see below for more details).

We performed pairwise comparisons of the original image (without holes) and each of the inpainting results. The smaller the difference between the two images, the more closely the inpainting result matches the original.

These two metrics rely on statistical values that, even if they are good for measuring the amount of artifacts introduced, do not consider if these artifacts will be obvious to a human observer. Therefore, we also considered another metric, proposed by Ardis and colleagues [Ard10a] that relates the visual saliency map of an image with its perceived quality, the Average Squared Visual Salience (ASVS). For computing the metric's results we considered the bottom-up visual saliency model, Graph-Based Visual Saliency (GBVS) [Har06a].

Color Layout [*CL*]: The CL metric examines the differences in the distribution of color in YUV space between the images to compare:

$$CL = \sqrt{\sum_{i\in Y}\alpha_i(Y_i - Y_i')^2} + \sqrt{\sum_{i\in U}\beta_i(U_i - U_i')^2} + \sqrt{\sum_{i\in V}\gamma_i(V_i - V_i')^2} \quad (5)$$

where the ith coefficient of each channel is denoted by Y_i , U_i , V_i. The weights represented by α, β and γ are inversely proportional to the coefficient scan order.

Edge Histogram [*EH*]: The EH descriptor is capable of capturing the spatial distribution of edges in an image via a combination of 5-bin normalized histograms. To generate each histogram, the image is segmented in 4x4 pixel patches and the intensity component Y in the YUV color space is used to extract and classify the edges putting them into the 5-bins (vertical, horizontal, both diagonals and non-directional):

$$EH(I_O, I_R) = \|EH(I_O) - EH(I_R)\|_1 \quad (6)$$

Average Squared Visual Salience [*ASVS*]: ASVS is a non-reference metric that focuses on the impact that introduction of artifacts in the inpainted area causes in the viewer attention:

$$ASVS(I) = \frac{1}{|\Omega|}\sum_{\Omega}(S'_{I_R}(p))^2 \quad (7)$$

where $S'_{I_R}(p)$ is the saliency corresponding to a pixel in the target area of the inpainted result.

4.2 Results of the Metrics

The values given by the three image quality metrics for the two different algorithms, averaged over the 16 images and all patch sizes, can be seen in Table 1. Since all metrics give either a value for dissimilarity between two images or a value for impact of artifacts, a higher value represent worse results. As can be seen in the table, two of the three metrics agree that introducing the amoeba improved the image quality. Moreover, the two metrics that indicate better performance for the Amoeba algorithm are the metrics that more directly relate to our aim of removing visually disruptive intrusion artifacts in the inpainted results.

	Criminisi	**Amoeba**
ASVS	0.278	**0.260**
CL	**22.431**	23.448
EH	36.103	**30.776**

Table 1: Results for the Criminisi and Amoeba algorithms averaged over the 16 images and all radii. The metrics' results indicate that the Amoeba improves the quality of the results in several complementary aspects

The results are a bit more nuanced when one looks at the individual images and patch sizes (see Table 2). Here, we will examine the results for the Criminisi algorithm first, then the amoeba, and finally we will compare the two.

The Criminisi Algorithm Results: The first interesting result, as can be seen in the table, relates to the Criminisi algorithm. The radius that yielded the best result is rarely one the "standard" radii (2, 3, 4 or 5). Specifically, ASVS, EH, and CL metrics found the typical radii to be best just for 2, 3, and 4 of the 16 images, respectively. In other words, in 81% of the analyses, the best size was not in the typical range. Furthermore, for no image do all three metrics agree that the best radii is in the typical range and for only 1 image do two metrics agree the typical range is best. In short, the best radius for the Criminisi algorithm would never have been tested in previous work! Note that this explicitly means that the quality of the Criminisi reconstructions found in this paper will be of better quality than is typically expected from this algorithm.

The second interesting finding for the Criminisi algorithm is that there is no clear way of predicting which patch size is best. Changing the patch radius even by one often produces radically different image qualities. Likewise, patches with very different radii often have nearly identical scores. Critically, images with similar dimensions (like *tiger* and *twobirds*) show totally different trends for the optimal patch size, suggesting that different features are selected and the decision is not solely dependent on the image size. All these observations indicate that the only way of finding the best fixed patch size for the Criminisi algorithm is brute force, with the consequent exponential increase of computational time.

It is also interesting to note that, for Criminisi, the three metrics never agreed on what the best radius was. In fact, on only 5 of the 16 images did two metrics agreed. Clearly the metrics were focusing on different features.

	Size		Best radii for Criminisi			Best radii for Amoeba		
	W	H	ASVS	CL	EH	ASVS	CL	EH
bicycle	*460*	*300*	**10** (0.296)	**3** (15.71)	**9** (18.71)	**5** (0.285)	**4** (17.164)	**11** (14.623)
bungee	*206*	*308*	**4** (0.301)	**8** (29.72)	**5** (38.41)	**19** (0.256)	**20** (29.605)	**19** (35.064)
butterfly	*1024*	*700*	**20** (0.186)	**1** (17.63)	**20** (21.75)	**16** (0.172)	**16** (19.176)	**18** (16.654)
butterfly2	*615*	*422*	**1** (0.369)	**1** (11.66)	**20** (6.33)	**5** (0.439)	**7** (15.536)	**19** (7.271)
car	*500*	*375*	**17** (0.190)	**3** (8.06)	**3** (8.34)	**8** (0.191)	**2** (8.057)	**3** (10.439)
cat	*1024*	*683*	**18** (0.163)	**9** (36.24)	**15** (42.74)	**3** (0.191)	**1** (36.478)	**14** (35.484)
colosseum	*512*	*340*	**20** (0.027)	**2** (16.82)	**9** (34.37)	**14** (0.027)	**19** (16.662)	**19** (30.317)
eagle	*600*	*402*	**20** (0.142)	**2** (20.25)	**17** (27.26)	**7** (0.132)	**1** (22.165)	**20** (19.914)
glasses	*500*	*395*	**9** (0.233)	**6** (28.48)	**17** (80.02)	**2** (0.274)	**4** (35.712)	**3** (57.846)
mochizuki	*574*	*346*	**18** (0.270)	**1** (17.78)	**19** (19.89)	**9** (0.263)	**1** (19.714)	**17** (16.706)
mountains	*512*	*683*	**9** (0.440)	**17** (14.99)	**15** (8.32)	**17** (0.416)	**15** (14.489)	**4** (10.428)
penguins	*615*	*461*	**13** (0.217)	**15** (13.24)	**19** (18.96)	**19** (0.232)	**8** (18.316)	**11** (18.801)
pigeons	*800*	*600*	**7** (0.223)	**19** (11.07)	**5** (15.61)	**3** (0.218)	**12** (10.840)	**2** (18.696)
soccer	*500*	*356*	**13** (0.116)	**19** (32.96)	**7** (56.33)	**16** (0.085)	**4** (35.495)	**6** (49.288)
tiger	*600*	*437*	**12** (0.091)	**20** (13.70)	**20** (19.75)	**20** (0.099)	**5** (15.951)	**14** (19.743)
twobirds	*600*	*450*	**5** (0.434)	**1** (31.68)	**16** (39.77)	**12** (0.303)	**1** (36.107)	**12** (27.267)

Table 2: Best values according to the metrics. The bold numbers indicate the radii that produced the best results for the Criminisi and Amoeba algorithms. Numbers in parenthesis reflect the metrics' scores. As discussed in the text, it seems that our approach provides similar or better image quality in terms of edges and perceptual saliency

A closer examination of the different preferences provides some useful insights. The CL metric prefers small patches (for 8 of the 16 images, the best size was a radius of 3 or smaller). For 5 images, CL preferred a large patch (15 or higher). A closer examination of the images suggests that this difference is being driven by the image features. The large patch size is preferred when the target region is inside a mostly homogeneous region (such as for *tiger* or *mountains*) and small when the target region has lots of texture or edges (such as *eagle* or *colosseum*). The EH metric, on the other hand, strongly prefers large patches (for 10 images the best radius was 15 or higher), and only once prefers small patches. This sole case where EH preferred a small patch was for an image (*car*) that had target regions with a few dominant edges at different spatial scales. The ASVS, which focused on visual saliency, preferred medium (9 images) or large (6 images) patches, and only once preferred small images.

The Amoeba Algorithm Results: As with Criminisi, the standard sized patches were not chosen very often: in 73% of the analyses, the best size was not in the typical range! Likewise, the three metrics never agreed on what the best patch size was. For only four of the images did two of the three metrics agree on the best patch size.

The preferences of the individual metrics was also similar to that of the Criminisi Algorithm. Specifically, the CL metric seems to prefer small patch sizes (for 5 of images the patch size was 3 or smaller, and for 8 o the images it was 4 or smaller). The EH metric seems to prefer large patches (for 6 images the best radius was 15 or higher; for 8 images it was 14 or higher). Smaller patches did, however, performed better than in the Criminisi algorithm. The ASVS preferred middle and large patches (6 at 15 or more; and only 3 with a radius of 3 or less).

Comparing Algorithms: Despite the same general trends for the preferences of the three metrics, the specific patch size that was seen as best in the two algorithm was very different. Only in three of the 48 analyses (16 images for 3 metrics) did a metric favor the same patch size for the same image in both algorithms. It is clear that the amoeba drastically altered the image reconstruction. As would be expected from the above discussion, there are few cases where either algorithm is a clear winner in terms of the computational metrics, as documented in Table 2. For 9 of the 16 images the ASVS felt that the amoeba outperformed the Criminisi algorithm and in 2 occasions the scores were tied. The EH metric felt that the amoeba outperformed the Criminisi algorithm in 12 of the 16 images. The CL metric, on the other hand, felt that the amoeba outperformed the Criminisi algorithm on a mere 5 images.

4.3 Face Validity

Given that the metrics rarely agree with each other, it is clear that we cannot rely on them too much without additional information about their relationship to perception. The relationship between the metrics and perception can be seen a bit more clearly in Figure 4 which shows the best image for both algorithms for each algorithm. For the image *tiger*, for example, two of the metrics rated Criminisi as better. A closer look at the images, however, shows that for each metric, our version had fewer intrusion artifacts and abrupt discontinuities. The image chosen by EH for our algorithm is clearly the most natural reconstruction. For the image *soccer*, only

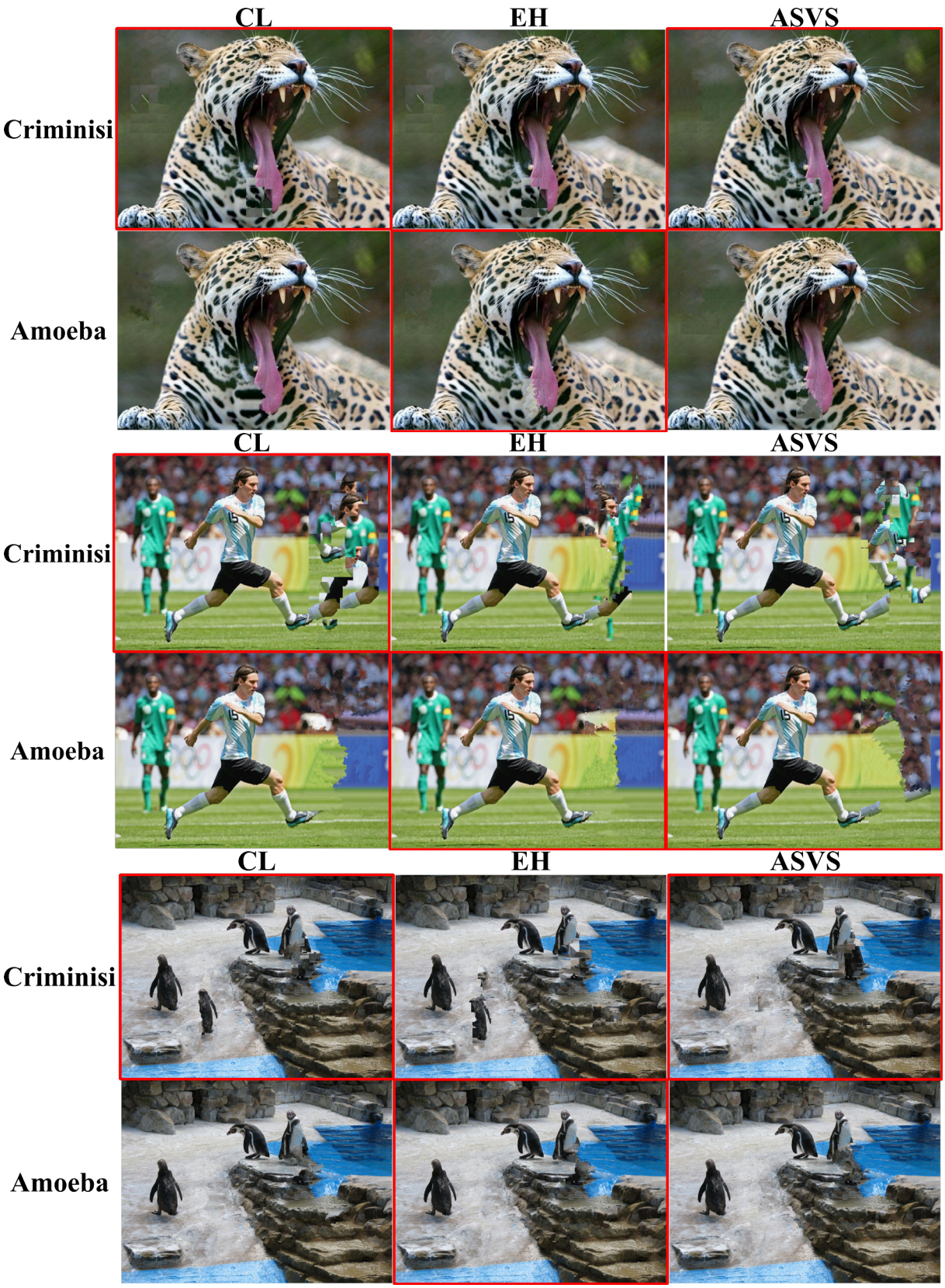


Figure 4: The best reconstruction according to the three metrics for the two algorithms. Each metric's preferred algorithm is highlighted with a red frame. As seen in the images the Amoeba algorithm results have visually fewer intrusions. For reference, the first row shows the original images without holes

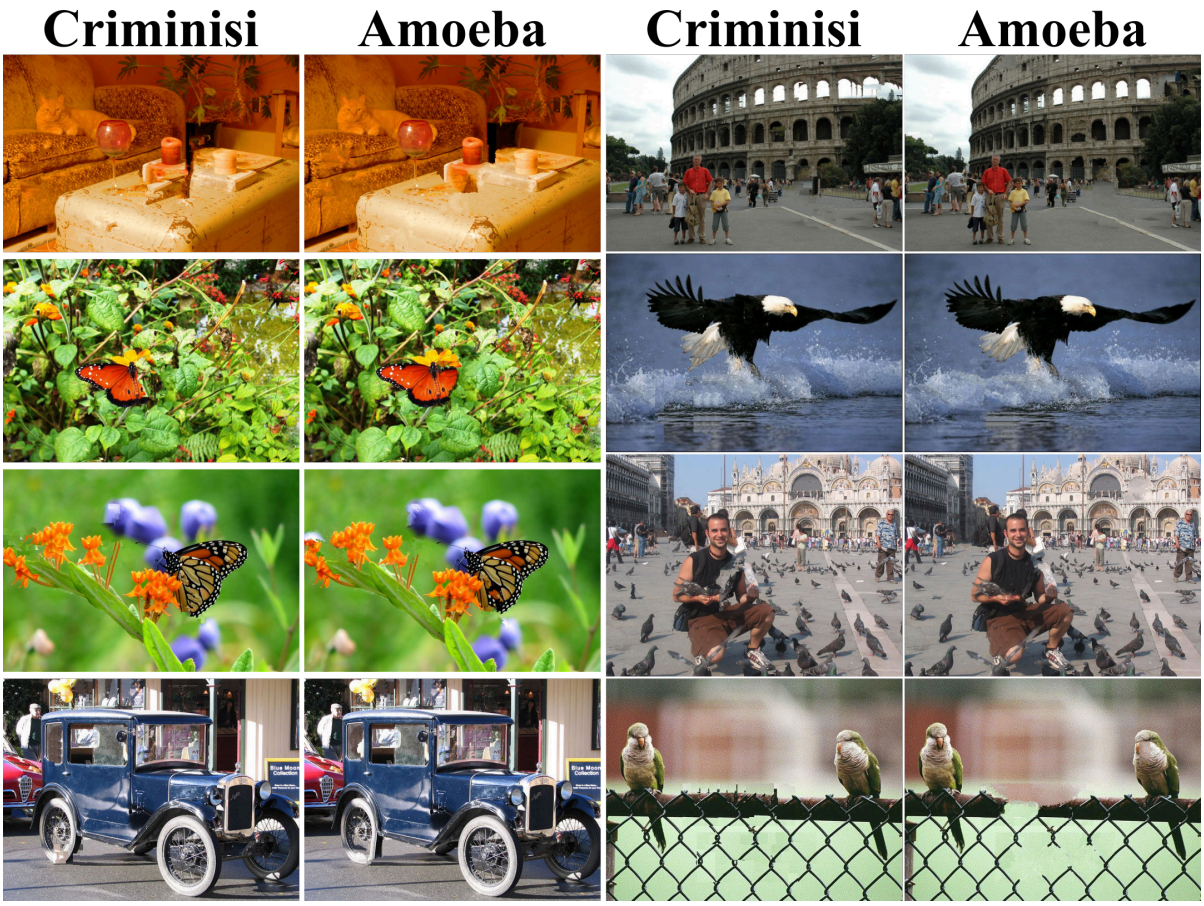


Figure 5: Subjectively selected best reconstructions for the two algorithms

CL preferred Criminisi. Yet clearly all of the chosen images for our algorithm contain fewer intrusion artifacts, and the results are definitely more perceptually plausible. It seems that the EH and to some degree the ASVS metrics are better at predicting the level of visual salience of disruptive intrusion artifacts. Finally, the image *penguins* two metrics preferred the results from Criminisi algorithm. Here again, we would argue that a closer look at the image reveals disconcerting artifacts in our reconstructions.

Since clearly the metrics are not very good at predicting human performance, with the possible exception of EH, it is possible that the best images chosen by the metrics, as seen in Figure 4, might not the best set of image to compare the two pipelines. Therefore, we examined the entire set of results (all 640 image) and presented them to several observers to choose the subjectively best image. A representative sample of the perceptually best image can be seen in Figure 5. Although clearly both algorithms can at times produce good reconstructions, it seems clear that the amoeba version has fewer intrusion artifacts and fewer disruptive or abrupt completions.

5 CONCLUSIONS

We extended the method and analysis of Criminisi et al. [Cri04a] in two ways. First, we showed that the best sized rectangular patches for the original, unmodified Criminisi algorithm are almost always well outside the range usually tested. Second, we demonstrated that growing a non-rectangular patch in the source area allows the algorithm to copy only pixels that most closely resemble the structure near the target pixel greatly reduces the chance of inserting artifacts into the target area and consequently improves image quality. This can be seen clearly in the figures and was confirmed by the image quality statistics. The amoeba we proposed has considerably reduced the occurrence of disembodied, partially completed, oddly placed structures. Given that this modification alters the size and shape of the patch itself, its effects are orthogonal to nearly all the other modifications to the method of Criminisi et al. made by other researchers and as such can be combined with them. Of course, matching and copying techniques that explicitly require a square patch will require modification to work with the new amoeba patches. Future work could focus on which of the many modules for each of the different stages in Criminisi's pipeline works optimally. Future work could also examine using the amoeba for the target patch as well. Finally, it is likely that amoebae can conceivably be used in any technique that employs patches to process, synthesize, or analyze images.

6 ACKNOWLEDGEMENTS

The authors would like to thank Stephan Guthe and Maximilian Mühle for their suggestions and valuable comments.

7 REFERENCES

[Ad17a] R.D. Adam, P. Peter, and J. Weickert. Denosing by inpainting. In *Proc. SSVM '17*, 121–132, 2017.

[Ard10a] P.A. Ardis, C.M. Brown, and A. Singhal. Inpainting quality assessment. *Journal of Electronic Imaging*, 19(1):011002–011002–7, 2010.

[Ber03a] M. Bertalmió, L. Vese, G. Sapiro, and S. Osher. Simultaneous structure and texture image inpainting. *IEEE Trans. Image Process.*, 12(8):882–889, 2003.

[Ber00a] M. Bertalmió, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proc. SIGGRAPH '00*, 417–424, 2000.

[Buy15a] P. Buyssens, M. Daisy, D. Tschumperlé, and O. Lézoray. Exemplar-based inpainting: Technical review and new heuristics for better geometric reconstructions. *IEEE Trans. Image Process.*, 24(6):1809–1824, 2015.

[Cai17a] N. Cai, Z. Su, Z. Lin, H. Wang, Z. Yang, and B.W.-K. Ling. Blind inpainting using the fully convolutional neural network. *The Visual Computer*, 33(2):249–261, 2017.

[Cas11a] S. Castillo, T. Judd, and D. Gutierrez. Using eye-tracking to assess different image retargeting methods. In *Proc. APGV '11*, 7–14, 2011.

[Cri04a] A. Criminisi, P. Perez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Trans. Image Process.*, 13(9):1200–1212, 2004.

[Ef99a] A.A. Efros and T.K. Leung. Texture synthesis by non-parametric sampling. In *Proc. ICCV*, vol. 2, 1033–1038, 1999.

[Gi79a] James J. Gibson. *The ecological approach to visual perception*. Boston: Houghton Mifflin, 1979.

[Gui14a] C. Guillemot and O. Le Meur. Image inpainting : Overview and recent advances. *IEEE Signal Process. Mag.*, 31(1):127–144, 2014.

[Har06a] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *Proc. NIPS*, pp. 545–552, 2006.

[Kat01a] E. Kasutani and A. Yamada. The mpeg-7 color layout descriptor: a compact image feature description for high-speed image/video segment retrieval. In *Proc. ICIP '01*, vol. 1, 674–677, 2001.

[Kom07a] N. Komodakis and G. Tziritas. Image completion using efficient belief propagation via priority scheduling and dynamic pruning. *IEEE Trans. Image Process.*, 16(11):2649–2661, 2007.

[Lem13a] O. Le Meur, M. Ebdelli, and C. Guillemot. Hierarchical super-resolution-based inpainting. *IEEE Trans. Image Process.*, 22(10):3779–3790, 2013.

[Lem11a] O. Le Meur, J. Gautier, and C. Guillemot. Examplar-based inpainting based on local geometry. In *Proc. ICIP*, 3401–3404, 2011.

[Lee12a] J. Lee, D. K. Lee, and R. H. Park. Robust exemplar-based inpainting algorithm using region segmentation. *IEEE Trans. Consum. Electron.*, 58(2):553–561, 2012.

[Ler07a] R. Lerallut, É. Decencière, and F. Meyer. Image filtering using morphological amoebas. *Image and Vision Computing*, 25(4):395–404, 2007.

[Liu08a] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W.T. Freeman. Sift flow: Dense correspondence across different scenes. In *Proc. ECCV*, 28–42, 2008.

[Mai09] M. Mainberger and J. Weickert. Edge-based image compression with homogeneous diffusion. In *Proc. CAIP*, 476–483, 2009.

[Man01a] B. S. Manjunath, J. R. Ohm, V. V. Vasudevan, and A. Yamada. Color and texture descriptors. *IEEE Trans. Circuits Syst. Video Technol.*, 11(6):703–715, 2001.

[Mas98a] S. Masnou and J.M. Morel. Level lines based disocclusion. In *Proc. ICIP*, vol. 3, 259–263, 1998.

[Ngu13a] H. M. Nguyen, B. C. Wünsche, P. Delmas and C. Lutteroth, Parameter optimisation for texture completion. In *Proc. IVCNZ '13*, 226–230, 2013.

[Rub10a] M. Rubinstein, D. Gutierrez, O. Sorkine, and A. Shamir. A comparative study of image retargeting. *ACM Trans. Graph.*, 29(6):160:1–160:10, 2010.

[Wu09a] J.Y. Wu and Q.Q. Ruan. A novel exemplar-based image completion model. *J. of Information Science and Engineering*, 25:481–497, 2009.

[Xi13a] X. Xi, F. Wang, and Y. Liu. Improved Criminisi algorithm based on a new priority function with the gray entropy. *Proc. CIS*, 214–218, 2013.

[Xu10a] Z. Xu and J. Sun. Image inpainting by patch propagation using patch sparsity. *IEEE Trans. Image Process.*, 19(5):1153–1165, 2010.

ISSN 1213-6972 Journal of WSCG http://www.wscg.eu/ Vol.26, 2018, No.2

Improving Multiple-Image Super-Resolution for Mobile Devices through Image Alignment Selection

Neil Patrick Del Gallego
De La Salle University
Taft Avenue, Malate
1004 Metro Manila,
Philippines
neil.delgallego@dlsu.edu.ph

Joel Ilao
De La Salle University
Taft Avenue, Malate
1004 Metro Manila,
Philippines
joel.ilao@delasalle.ph

ABSTRACT

Multiple-image super-resolution (MISR) attempts to recover a high-resolution (HR) image from a set of low-resolution (LR) images. In this paper, we present a mobile MISR tailored to work for a wide range of mobile devices. Our technique aims to address misalignment issues from a previous work and further enhance the quality of HR images produced. The proposed architecture is used to implement a prototype application that is freely available at Google Play Store, titled Eagle-Eye HD Camera. The system is divided into the following modules: Input Module, Edge Detection Module, Image Selection Module, Image Alignment Module, Alignment Selection Module and Image Fusion Module.

We assessed the quality of HR images produced by our mobile MISR, through an online survey, as well as compare it with other related SR works. Performance time was also measured. A total of 114 respondents have participated in the survey, where majority of respondents preferred our approach. Our approach is observed to be comparable with other SR works in terms of visual quality and performance time, and guaranteed to work in a mobile environment.

Keywords

Super-resolution, Mobile Devices, Mean Fusion, Image Alignment

1 INTRODUCTION

Multiple-image super-resolution (MISR) attempts to recover a high-resolution (HR) image from a set of low-resolution (LR) images. Figure 1 shows HR images produced by our proposed mobile multiple-image super-resolution (MMISR) system for mobile devices. To the best of the authors' knowledge, there are limited studies and implementations of super-resolution on mobile devices, presumably because of its high time and space complexity. However, mobile devices are already capable of implementing a mobile MISR system, provided that the system makes efficient use of its hardware resources. Mobile device manufacturers such as ASUS[1], Vivo[2], and OPPO[3] include a camera feature that mimics an MISR technique to capture HR images. Similar MMISR studies were observed from [Chu13], [ZC14], [ZWZ13], and [DS15].

Images obtained from mobile devices may be modeled as having undergone a series of noise, downsampling and motion blur, which is similar to the image degradation model proposed in [MPSC09]. The goal of any MMISR system is to reverse these degradation effects.

An MISR technique can be divided into the following steps [NM14]: denoising, deblurring or image selection, alignment, upsampling and image fusion. In our implementation, the steps are performed in a sequential manner and memory is being managed by our matrix pool discussed in Section 4.1. The contributions of this study can be summarized below:

[1] How to Shoot Super Resolution on ZenFone 4: https://youtu.be/o3DFhZxzwtk

[2] Vivo V7. 64MP Ultra HD Photos: https://www.vivo.com/product/en/product/v7

[3] OPPO Pure Image, Ultra HD: https://www.oppo.com/en/technology/pure-image

https://doi.org/10.24132/JWSCG.2018.26.2.7

Figure 1: Sample HR images produced by the SR system with noticeable improvement. A: Bicubic interpolation. B: Proposed SR method.

1. We implemented and published a free prototype application, titled Eagle-Eye HD Camera[4].
2. We presented a revised architecture for performing MMISR, that aims to address misalignment issues from the work of [DGI17].
3. We compared the quality of resulting HR images with the Bicubic baseline, and other related SR works. We also conducted an online survey to assess our HR images, where 114 respondents have participated. The survey shows promising results to further improve our work.
4. We developed a method for objectively assessing the quality of aligned images. The MMISR prototype performs alignment selection based on this assessment method.

Our paper is organized as follows: We review recent work in SR for mobile devices in Section 2. Specifically, we discuss limitations of [DGI17] and how we address these in Section 3. Lastly, we discuss our results in Section 5 and conclusions in Section 6.

2 RELATED WORK

Image super-resolution is still needed despite the advances in hardware such as the introduction of high-definition (HD) displays. Some high-end mobile devices introduced as of 2018 have a resolution of 1440 $\times$ 2560 known as quad-HD displays and camera resolution size may go as large as 40MP [YSL+16][5]. However, development of MMISR seems to be limited due to its computational cost. Mobile devices are typically equipped with burst mode capture which can be utilized to perform MISR, provided that the images captured have substantially different pixel values. This is proven in the work of [DGI17], that there are substantial differences from images captured using the burst mode of the camera [DGI17]. [Chu13] proved that multiple images captured from mobile devices result in small motions due to high frame rates, where an affine flow model is suitable for aligning the images. A joint image alignment and deblurring approach was also proposed by [ZC14].

The MMISR system developed by [DGI17] mostly works when the user captures images steadily, or the subject have adequate lighting. Limitations and issues were observed, which are summarized below:

1. Limitation **L1**: Images become misaligned whenever images are captured with shaky hands. While the system has some tolerance for aligning images with slight angular changes, it can only work ideally when images are captured steadily. This scenario may not be practical for most end-users. An affine and perspective transformation estimation was used to align the images which proves to be an insufficient approach as discussed in Section 2.1.
2. Limitation **L2**: Inadequate lighting and changes in exposure values also affects the alignment.
3. Limitation **L3**: Using a mean fusion approach may smoothen the pixel values. While mean fusion can be effective for removing noise, it also causes some high-frequency details to be lost. The problem is that the system does not employ any regularization-based methods as observed from related approaches [MPSC09, NMG01, LHG+10, PC12, YZS12].

2.1 Misalignment Issues

The MMISR system of [DGI17] implements Affine Transformation Estimation (ATE) [Ho15] and Perspective Transform Estimation (PTE) [Ho15] sequentially. However, their technique can produce misaligned images and causes unwanted artifacts to appear in the

[4] Eagle-Eye HD Camera: `https://play.google.com/store/apps/details?id=neildg.com.eagleeyesr`

[5] Huawei P20 Pro: `https://consumer.huawei.com/en/phones/p20-pro`

HR image. Figure 2 exhibits blurring and distortion of texts due to incorrect transform estimation. Figure 3 shows a low-light image example taken with varying exposure values (-3, 0, +3 respectively). In this example, a warping distortion occurred due to lack of image detail or varying exposure value. There is not enough reliable keypoints available for correctly estimating the perspective transformation. Figure 4 introduces a ghosting effect. ATE and PTE is performed globally on the whole image. Hence, it cannot handle localized transformation on image regions (Image B and C).

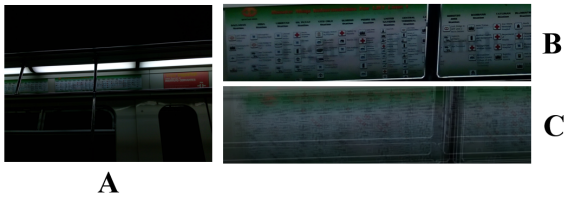


Figure 2: Misalignment example due to incorrect transform estimation. A: one of the LR image sequences. B: zoomed region on one of the images. C: zoomed region on an image with misalignment.

Figure 3: Low-light images with varying exposure values (EV) are prone to misalignments. A: Low-light LR images with -3, 0, +3 EV. B: zoomed region on one of the images. C: zoomed region on an image with warping distortion.

Figure 4: Misalignment example on a scenery image. A: one of the LR image sequences. B: zoomed region on one of the images. C: zoomed region on an image with ghosting effect.

2.2 Loss of Detail after Mean Fusion

As mentioned in **L3**, it is observed from the method of [DGI17] that the nature of the mean fusion process smoothens out some of the high-frequency details of the images as illustrated in Figure 5.

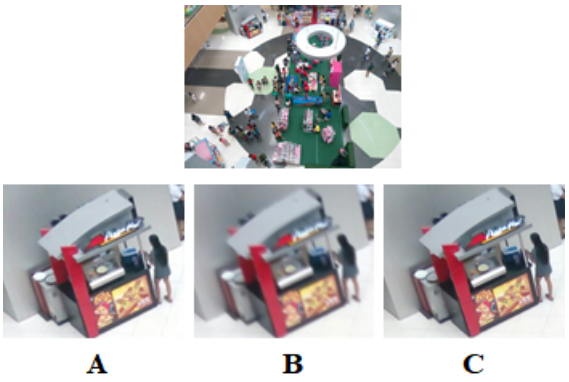


Figure 5: A: cubic interpolation. B: SR method by [DGI17]. C: ground-truth. Loss of detail is exhibited in result of B after performing mean fusion.

3 ADDRESSING THE LIMITATIONS

This section discusses our proposed approaches that aim to address **L1**, **L2** and **L3**. Our proposed MMISR system selects well-aligned images from the outputs of two alignment algorithms, PTE and MTB (Median Threshold Bitmap alignment [War03]). The choice of these alignment techniques are influenced by the following factors:

1. PTE can easily be performed on a mobile device because of its low computational cost and fast processing time (see performance time discussion). Images captured from mobile devices typically have a resolution of 8MP or more, and these techniques can handle images with large resolution.

2. MTB is observably the fastest and most reliable technique for aligning images captured on a mobile device [War03].

To verify the quality of image alignment algorithms, an experiment was performed that compared ATE, PTE and MTB through a fitness score. It is observed that misaligned images introduce additional edges when merged with the reference image as shown in Figure 6. With this observation, and because there are no standard mesures for assessing how an image is well-aligned, we propose a technique that measures the density of edges through Sobel derivatives [Sob68]. Correctly aligned images should not introduce additional edges from the reference image. To detect this observation, the following steps are performed:

1. Let L_0 be the first reference LR image, $\{A_1 ... A_N\}$ are aligned image sets produced by image alignment technique A.

2. Count the non-zero elements of the edge image for L_0 to produce an integer measure, e_0.

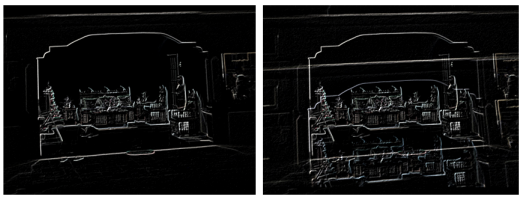


Figure 6: Misaligned images introduce additional edges. A: reference image edge. B: misaligned image merged with the reference image.

3. For $a \in \{A_1 ... A_N\}$, add a with L_0 to produce the edge image, as seen in Image B in Figure 6. Let this be $\{\bar{A_1} ... \bar{A_N}\}$

4. Count the non-zero elements of the edge images for $\{\bar{A_1} ... \bar{A_N}\}$. Let this be $\{\bar{e_1} ... \bar{e_N}\}$.

5. Compute for the integer measures, $\{\varepsilon_i ... \varepsilon_N\}$ by simply subtracting $\bar{e_i}$ to e_0, where i is $1...N$. Label this as *SobelMeasure*.

$\{\varepsilon_i ... \varepsilon_N\}$ refers to the corresponding *SobelMeasure* values of aligned image set, $\{A_1 ... A_N\}$. A low value indicates that minimal edges were introduced when attempting to combine the aligned images to the reference image.

Using the proposed technique, 33 image sets where gathered and aligned, where each image set consists of 10 images captured using burst mode. The resulting average *SobelMeasure* of aligned image sets are visualized in Figure 7. Out of 33 image sets tested, PTE works best on 28 image sets, ATE works best for 3 image sets, and MTB alignment works best for 2 image sets. PTE is the ideal image alignment technique for images taken from mobile devices. However, MTB sometimes aligns an image sequence better than PTE. This is where we propose an alignment selection technique. We select an aligned image by selecting the lesser difference in *SobelMeasure*. Suppose P is the aligned image using PTE, and M is the aligned image using MTB, for some $\{L_0 ... L_N\}$ image sequence. If *SobelMeasure* of $P \leq M$, then P will be selected as the aligned image for that image sequence. Otherwise, M will be selected. This is demonstrated in Figure 8 where artifacts are severely reduced in the final image.

For addressing **L3**, an L_1-norm minimization approach proposed by [FREM04] may be applied or other regularization-based approaches in recent works [LHG+10, PC12, YZS12]. However, such a technique may result in a huge computational time for a mobile device due to its iterative nature. Based from this assumption, we simply applied a sharpening operation, unsharp masking, to individual LR images which proves to be effective in preserving edges while also removing noise as observed in Figure 9. Edges as

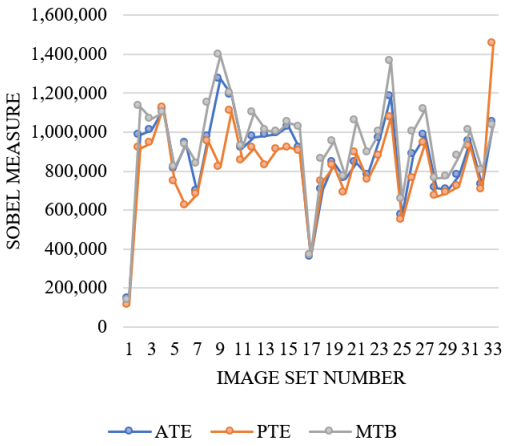


Figure 7: *SobelMeasure* values from 33 image sets visualized as a line chart. A lower value indicates that the image sequences are more well-aligned to its reference image. PTE has the lowest average *SobelMeasure* of **818,056**.

Figure 8: A: alignment using Perspective Transform Estimation. B: Best Alignment Technique. Aligning images with varying exposure values is a clear limitation of the MMISR system. This results in severe warping distortion only if (A) was applied. Misalignment and warping distortion is reduced if (B) was applied.

well as noise gets amplified but performing a mean fusion to combine all unsharp masked LR images will create an image where edges are preserved while also minimizing noise.

4 OUR PROPOSED ARCHITECTURE

Our system architecture borrows from the architecture presented in [DGI17], but modified that image alignment technique to address **L1** and **L2** and applied unsharp masking to LR images to mitigate **L3**. The system architecture is shown in Figure 10.

The system accepts a set of LR images wherein the first LR image serves as the reference LR image. In the Edge Detection and Image Selection Module, the LR images undergo the same feature-selection scheme proposed in [DGI17]. The Image Selection Module produces a filtered set of LR images, $\{L_0 ... L_N\}$ where an Unsharp Masking operator is applied to the images, to

Figure 9: Unsharp masking illustration. A: LR image. B: LR image with unsharp masking applied. C: Resulting image after performing mean fusion. Observe that noise and other artifacts seen in B was suppressed, while having the edges preserved in C.

address **L3**. Unlike [DGI17], non-local means denoising [BCM11] can be applied optionally, to conserve computation time. L_0 is upsampled using bicubic interpolation, which becomes the initial HR image $\hat{H}$.

The subset $\{L_1...L_N\}$ undergoes PTE [Ho15], and MTB alignment [War03]. This produces warped images, $\{P_1...P_N\}$ for the PTE-aligned images and $\{M_1...M_N\}$ for MTB-aligned images.

The image sets $\{P_1...P_N\}$ and $\{M_1...M_N\}$ enter the Image Alignment Module where an image that introduces the least error in alignment will be chosen, which addresses **L1** and **L2**. The selected aligned image, $\{W_k\}_{k=1}^N$, can either be $\{P_k\}_{k=1}^N$ or $\{M_k\}_{k=1}^N$ respectively.

The images $\{W_1...W_N\}$ are upsampled using bicubic interpolation to produce $\{\hat{H}_1...\hat{H}_N\}$ as initial images to be mapped to the HR grid, by mean fusion. This produces the final HR image, H.

4.1 Matrix Pooling for Optimization

We discuss matrix pooling as an optimization technique that made our MMISR implementation possible. In our implementation, we prioritized memory management over computational time to minimize the chances of out of memory errors.

Matrix pooling is heavily inspired from the object pool software design pattern, but applied to matrices. N matrices of size $H \times W$ are pre-allocated at startup. Each module may request for M matrices where $M \leq N$ for processing. After a task has been performed, M matrices are released back to the pool of N matrices. Should $M > N$, then this returns a failure. Otherwise, the tasks in the system modules perform as is. Because of matrix pooling, memory can easily be managed and results in faster computational time, as the matrices are only instantiated during the start of an SR task, and destroyed when the SR task is completed.

5 RESULTS AND DISCUSSION

5.1 Assessment of HR Images

A preliminary survey was conducted to assess the quality of images. 114 respondents have participated. These images do not have any ground-truths as they were captured in a real-world scenario. The respondents where tasked to evaluate zoomed images. The survey is structured as follows: a total of 42 randomly selected image sets with HR images were requested to be evaluated by the respondents. There are 3 choices to choose from, **Method A**: a bicubic interpolated image, **Method B:** an HR image produced by the previous SR method [DGI17], and **Method C**: our proposed SR method. In the survey, an image thumbnail is provided followed by the 3 HR images zoomed in on a certain region, as illustrated in Figure 11. The image choices were randomized per question so that respondents will not discover any patterns in the choices.

The respondents choose one of the 3 image choices provided, followed by a confidence level rating from 1 to 5. This indicates the confidence and certainty of their chosen preferred image. A rating of 5 means that the respondent is very sure of his/her image choice. A rating of 1 means that the respondent had difficulties choosing his/her preferred image or their decision is split among the other image choices.

5.2 Preliminary Survey Results

The survey results are summarized in Table 1. The "Number of Majority Votes" column tallies the number of test images where a given technique is selected by majority. The "Average Confidence Level 5" column indicates the average percentage of Level 5 ratings for a given technique.

It is shown in Table 1 that Method C was selected as majority for 26 test images with Average Confidence Level 5 of 41.95%. It is followed by the Method A were it was selected as majority for 14 test images with Average Confidence Level 5 of 39.16%. Method B were only selected for 2 test images and Average Confidence Level 5 of 37.30%

Based from the results, it can be justified that Method C were preferred by respondents over Method A and B. Whenever the produced HR images from Method C are not preferable to respondents, the HR images produced by Method A were selected.

Figure 13 shows the best quality HR images preferred by respondents (refer to Figure 12 for the thumbnails).

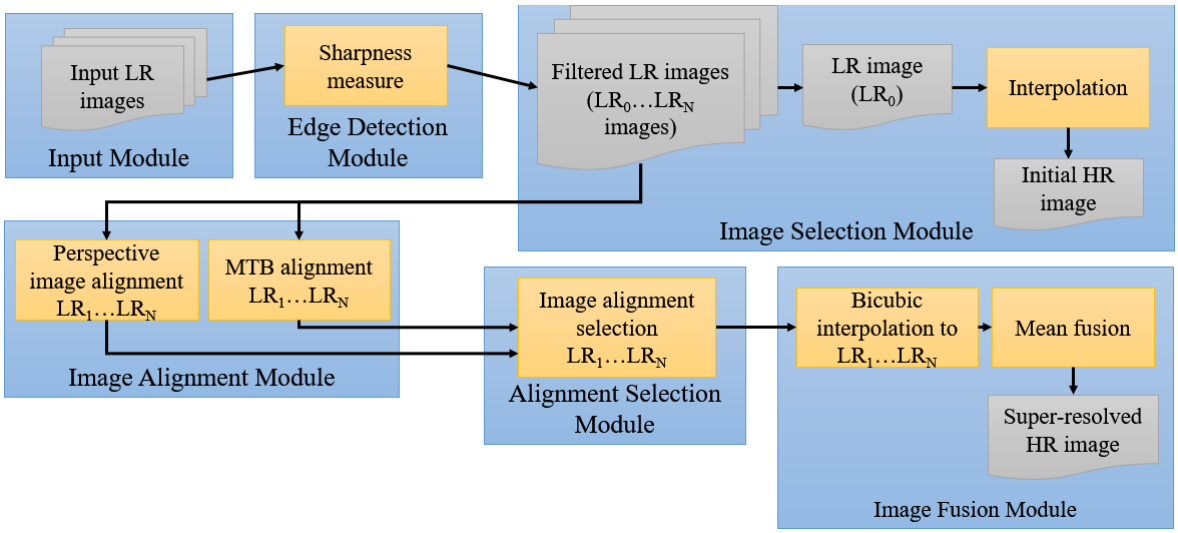


Figure 10: System architecture of our MMISR system.

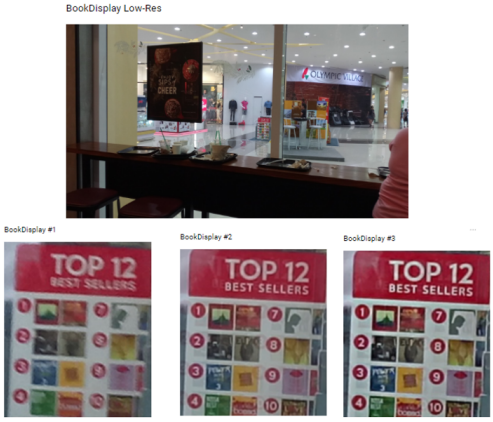


Figure 11: A snippet of the online survey. Image thumbnail is presented first followed by three image choices.

Table 1: Summary of survey results

Technique	**Number of Majority Votes**	**Average Confidence Level 5**
Method A	14 out of 42	39.16%
Method B	2 out of 42	37.30%
Method C	26 out of 42	41.95%

Image 1 received the highest number of votes (95.60%) for Method C, with a Level 5 Confidence of 69.30%. Image 5 has 90.40% of votes and also has the highest Level 5 Confidence Level Percentage, which is 72.80%. It can be observed from the best cases that users prefer clear texts and sharp details. Method C produced the clearest and most visible text than the other techniques in these image sets. Additional results are shown in Figure 18 and 19.

5.3 Performance Time

Processing time and space consumption were measured using a test device with 2.0 Ghz Octa-core processor, 4GB RAM, and 16MP rear camera. A total of 20 image sets were used and the processing time was averaged. The LR images have a size of 2992 $\times$ 5280 resolution, which is the default resolution size of the camera. The MMISR system produces 50MP HR images of 5980 $\times$ 10560. Figure 14 and 15 shows the average performance time and standard deviation respectively. The Image Alignment, Alignment Selection and Image Fusion module takes up at least 60 seconds to process. With this observation, the processing time of Image Alignment and Alignment Selection modules can be further reduced by having only 1 robust alignment technique implemented similar to how [ZC14] handled image alignment. Denoising has the highest standard deviation because the quantity of images selected by the Image Selection Module, varies across test sets. Each additional image also results in a huge increase in denoising time, which is why it was made an optional feature and only ideal for low-light images.

5.4 Comparison with Related SR Work

We compared our results to related SR works by using ten frames from the video provided in [FREM04]. The frames have a resolution of 49 $\times$ 57. In Figure 16, we compared our approach to the following: Bicubic baseline, [FREM04] because we want to compare its L_1 minimization approach with our simpler unsharp masking technique to address the "smoothening" effect of mean fusion, and [ZC14] because of its promising approach of joint alignment and deblurring. Using a scaling factor of 2, our MMISR method outperforms the Bicubic method and the method of [FREM04]. In terms of image quality and sharpness, the method of [ZC14] outperforms our MMISR. In terms of speed, MMISR

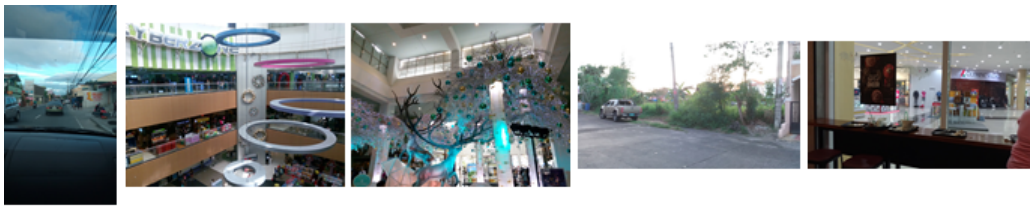


Figure 12: Thumbnails of the best quality HR images (Top 1 - 5), preferred by respondents.

Method	Image 1	Image 2	Image 3	Image 4	Image 5
A	2.60%	3.50%	7.90%	5.30%	9.60%
B	1.80%	2.60%	1.80%	4.40%	0.90%
C	**95.60%**	**93.90%**	**90.40%**	**90.40%**	**89.50%**

Figure 13: Best HR images preferred by respondents with the corresponding percentage of votes. 114 respondents have participated in the survey. Images 1 - 5 received above 89% of votes in Method C (our method), which effectively surpasses the bicubic performance (Method A) and the SR method of [DGI17]. It can be observed from the best cases that users prefer clear texts and sharp details.

is considerably faster than the method of [ZC14]. The performance time of the method of [ZC14] is **155.63 seconds** compared to MMISR which is less than **1 second**. The deblurring and denoising stage in the approach of [ZC14] were the most time-consuming. While the technique of [ZC14] is robust and can handle extremely blurred and noisy images, MMISR is considerably faster and more appropriate for mobile devices.

We also compared our results to the work of [KJ14], because they perform a specialized approach in image upsampling through self-learning. Additionally, we compared our results to the work of [KJ13] and [SLJT08]. The test images in the work of [KJ14] were re-captured from a computer screen, using a mobile device with a 16MP camera. This was performed so that the mobile camera settings such as ISO, exposure and shutter speed, affect the quality of the HR images and provide a fair analysis against the mentioned SR techniques. Figure 17 shows the results. Our proposed technique, produces comparable results as that of related single-image SR works. It can be observed that our technique produces clearer edges among other SR methods.

6 CONCLUSION AND FUTURE WORK

This research presents an improved framework for implementing a mobile multiple-image super-resolution system (MMISR) for mobile devices, by addressing the limitations observed in the implementation of [DGI17]. The system architecture is divided into the following modules: Input Module, Edge Detection Module, Image Selection Module, Image Alignment Module, Alignment Selection Module and Image Fusion Module.

Our results, based from the survey and analysis of the performance time, show a promising direction in MMISR research. Immediate steps needed to further improve and validate our system is to compare it with other state-of-the-art approaches, and apply some of the techniques seen in single-image SR works [TDSVG15,

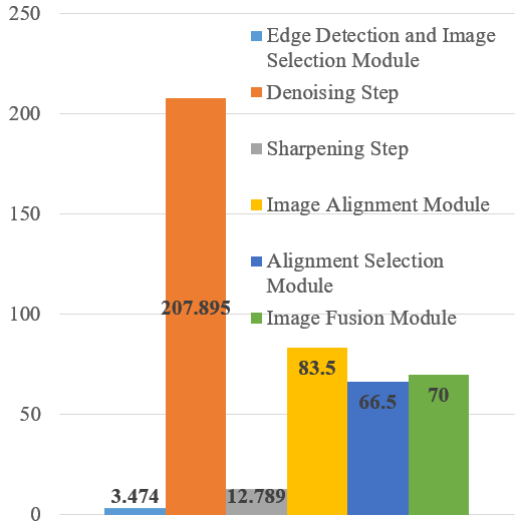


Figure 14: Average performance time of system modules in seconds. Denoising is an optional feature in the actual application due to its very long processing time. Performing image alignment, and image fusion are the heaviest in terms of processing time.

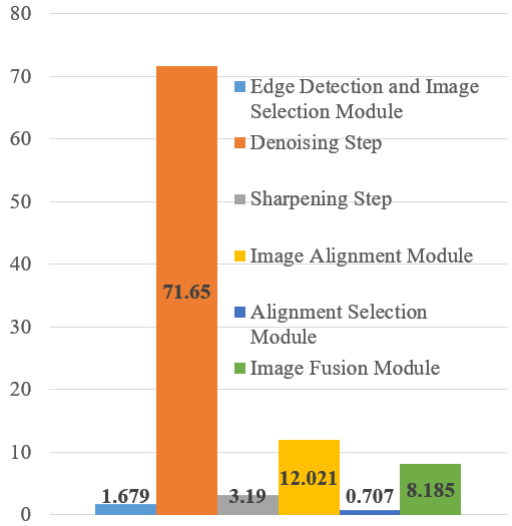


Figure 15: Standard deviation of system modules in seconds. Denoising has the highest standard deviation because the quantity of images selected by the Image Selection Module, varies across test sets.

DLHT16, TAe17]. We also plan to implement specialized approaches in image upsampling [RU90, NTP17].

Based from the preliminary survey conducted, results show that respondents prefer our approach. However, a more thorough analysis on the preferences of users must be performed [YMY14]. Using an Unsharp Masking operator to solve **L3** may introduce artificial artifacts in the images. Thus, it is recommendable that a specialized sharpening operation is performed such that it preserves the natural contours and composition of the images, which is an interesting approach in this paper [RIM17].

Figure 16: Comparison with related SR work. A: Bicubic image. B: Method of [FREM04]. C: Method of [ZC14]. D: Our method. While Method C contains more high-frequency details than Method D, Method D is considerably faster than Method C, but also has more detail than Method A and B.

A ACKNOWLEDGMENTS

The authors would like to acknowledge the Department of Science and Technology - Philippine Council for Industry, Energy and Emerging Technology (Project Grant No. 3897) and DLSU Science Foundation for funding this research.

The mobile image dataset used for this study is publicly available at Github: `https://github.com/NeilDG/EagleEyeDataset`

REFERENCES

[BCM11] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. Non-local means denoising. *Image Processing On Line*, 1, 2011.

[Chu13] Chung-Hua Chu. Super-resolution image reconstruction for mobile devices. *Multimedia Systems*, 19(4):315–337, July 2013.

[DGI17] Neil Patrick Del Gallego and Joel Ilao. Multiple-image super-resolution on mobile devices: an image warping approach. *EURASIP Journal on Image and Video Processing*, 2017(1):15, 2017.

[DLHT16] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):295–307, Feb 2016.

[DS15] M. Delbracio and G. Sapiro. Burst deblurring: Removing camera shake through fourier burst accumulation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2385–2393, June 2015.

[FREM04] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar. Fast and robust multiframe super resolution. *IEEE Transactions on Image Processing*, 13(10):1327–1344, Oct 2004.

[Ho15] Nghia Ho. Understanding opencv estimate rigid tranform, 2015.

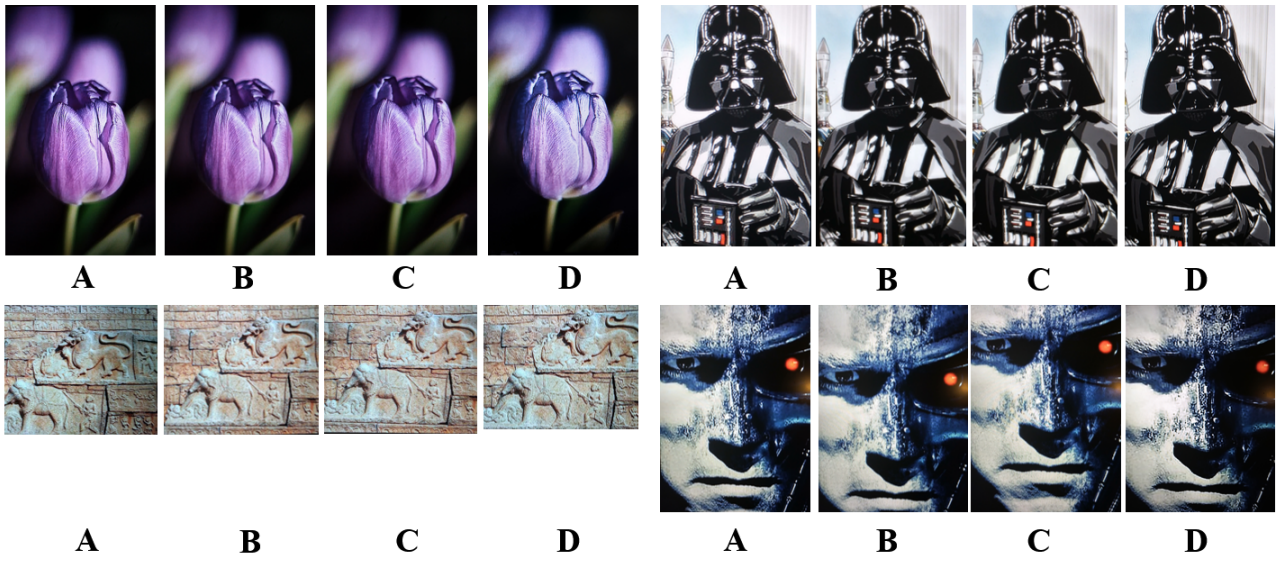


Figure 17: Comparison with related SR work. A: Method by [SLJT08]. B: Method by [KJ13]. C: Method by [KJ14]. D: Our method. Our method is observed to produce clearer edges.

Figure 18: Thumbnails of the best quality HR images (Top 6 - 10), preferred by respondents.

Method	Image 6	Image 7	Image 8	Image 9	Image 10
A	0.90%	0.0%	1.80%	3.50%	4.40%
B	9.60%	15.80%	18.40%	17.50%	16.70%
C	**89.50%**	**84.20%**	**79.80%**	**78.90%**	**78.90%**

Figure 19: Best HR images (Top 6 - 10) preferred by respondents with the corresponding percentage of votes. 114 respondents have participated in the survey. Method A: Bicubic image. B: SR method of [DGI17]. C: Our method.

[KJ13] Nilay Khatri and Manjunath V. Joshi. Image super-resolution: Use of self-learning and gabor prior. In Kyoung Mu Lee, Yasuyuki Matsushita, James M. Rehg, and Zhanyi Hu, editors, *Computer Vision – ACCV 2012*, pages 413–424, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[KJ14] N. Khatri and M.V Joshi. Efficient self-learning for single image upsampling. In *22nd International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG 2014)*, pages 1–8, 2014.

[LHG+10] Xuelong Li, Yanting Hu, Xinbo Gao, Dacheng Tao, and Beijia Ning. A multi-frame image super-resolution method. *Signal Processing*, 90(2):405 – 414, 2010.

[MPSC09] Dennis Mitzel, Thomas Pock, Thomas Schoenemann, and Daniel Cremers. Video super resolution using duality based tv-l1 optical flow. *Proceedings of the 31st DAGM Symposium on Pattern Recognition*, pages 432–441, 2009.

[NM14] Kamal Nasrollahi and Thomas B. Moeslund. Super-resolution: A comprehensive survey. *Machine Vision Applications*, 25(6):1423–1468, August 2014.

[NMG01] Nhat Nguyen, P. Milanfar, and G. Golub. A computationally efficient superresolution image reconstruction algorithm. *IEEE Transactions on Image Processing*, 10(4):573–583, Apr 2001.

[NTP17] Mattia Natali, Giulio Tagliafico, and Giuseppe Patan. Local up-sampling and morphological analysis of low-resolution magnetic resonance images. *Neurocomput.*, 265(C):42–56, November 2017.

[PC12] P. Purkait and B. Chanda. Super resolution image reconstruction through bregman iteration using morphologic regularization. *IEEE Transactions on Image Processing*, 21(9):4029–4039, Sept 2012.

[RIM17] Y. Romano, J. Isidoro, and P. Milanfar. Raisr: Rapid and accurate image super resolution. *IEEE Transactions on Computational Imaging*, 3(1):110–125, March 2017.

[RU90] S. P. Raya and J. K. Udupa. Shape-based interpolation of multidimensional objects. *IEEE Transactions on Medical Imaging*, 9(1):32–42, Mar 1990.

[SLJT08] Qi Shan, Zhaorong Li, Jiaya Jia, and Chi-Keung Tang. Fast image/video upsampling. *ACM Transactions on Graphics*, 27(5):153:1–153:7, December 2008.

[Sob68] Feldman G. Sobel, I. A 3x3 isotropic gradient operator for image processing. *Stanford Artificial Intelligence Project*, 1968.

[TAe17] R. Timofte, E. Agustsson, and L. V. Gool et.al. Ntire 2017 challenge on single image super-resolution: Methods and results. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1110–1121, July 2017.

[TDSVG15] Radu Timofte, Vincent De Smet, and Luc Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. *Computer Vision – ACCV 2014: 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part IV*, pages 111–126, 2015.

[War03] Greg Ward. Fast, robust image registration for compositing high dynamic range photographs from handheld exposures. *Journal of Graphics Tools*, 8:17–30, 2003.

[YMY14] Chih-Yuan Yang, Chao Ma, and Ming-Hsuan Yang. Single-image super-resolution: A benchmark. In *Proceedings of European Conference on Computer Vision*, 2014.

[YSL+16] Linwei Yue, Huanfeng Shen, Jie Li, Qiangqiang Yuan, Hongyan Zhang, and Liangpei Zhang. Image super-resolution: The techniques, applications, and future. *Signal Processing*, 128:389 – 408, 2016.

[YZS12] Q. Yuan, L. Zhang, and H. Shen. Multi-frame super-resolution employing a spatially weighted total variation model. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(3):379–392, March 2012.

[ZC14] H. Zhang and L. Carin. Multi-shot imaging: Joint alignment, deblurring, and resolution-enhancement. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2925–2932, June 2014.

[ZWZ13] H. Zhang, D. Wipf, and Y. Zhang. Multi-image blind deblurring using a coupled adaptive sparse prior. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1051–1058, June 2013.