

**The 15<sup>th</sup> International Conference in Central Europe on  
Computer Graphics, Visualization and Computer Vision 2007**

in co-operation with

**EUROGRAPHICS**

# **W S C G ' 2007**

University of West Bohemia  
Plzen  
Czech Republic

January 29 – February 1, 2007

## **Short Communications Proceedings **KK****

*Co-Chairs*

**Jarek Rossignac, Georgia Tech, USA**  
**Vaclav Skala, University of West Bohemia, Plzen, Czech Republic**

*Edited by*  
Vaclav Skala

***WSCG'2007 Short Communications Proceedings***

Editor-in-Chief: Vaclav Skala  
University of West Bohemia, Univerzitni 8, Box 314  
CZ 306 14 Plzen  
Czech Republic  
*skala@kiv.zcu.cz*

Managing Editor: Vaclav Skala

Author Service Department & Distribution:  
Vaclav Skala - UNION Agency  
Na Mazinách 9  
322 00 Plzen  
Czech Republic

Printed at the University of West Bohemia

Hardcopy: *ISBN 978-80-86943-02-2*

# WSCG 2007

## International Programme Committee

Alexa, Marc (Germany)  
Bartz, Dirk (Germany)  
Bekaert, Philippe (Belgium)  
Benes, Bedrich (U.S.A.)  
Bengtsson, Ewert (Sweden)  
Bieri, Hanspeter (Switzerland)  
Biri, Venceslas (France)  
Bouatouch, Kadi (France)  
Chen, Min (U.K.)  
Chrysanthou, Yiorgos (Cyprus)  
Coquillart, Sabine (France)  
Davis, Larry (U.S.A.)  
Deussen, Oliver (Germany)  
du Buf, Hans (Portugal)  
Ferguson, Stuart (U.K.)  
Ferko, Andrej (Slovakia)  
Goebel, Martin (Germany)  
Groeller, Eduard (Austria)  
Hauser, Helwig (Austria)  
Hege, Hans-Christian (Germany)  
Jansen, Frederik (The Netherlands)  
Klosowski, James T. (U.S.A.)  
Kobbelt, Leif (Germany)  
Kruijff, Ernst (Austria)  
Magnor, Marcus (Germany)  
Midkiff, Sam (U.S.A.)  
Myszkowski, Karol (Germany)  
Pasko, Alexander (Japan)  
Peroche, Bernard (France)  
Puppo, Enrico (Italy)  
Purgathofer, Werner (Austria)  
Rauterberg, Matthias (The Netherlands)  
Rokita, Przemyslaw (Poland)  
Rossignac, Jarek (U.S.A.)  
Rudomin, Isaac (Mexico)  
Sbert, Mateu (Spain)  
Schilling, Andreas (Germany)  
Schumann, Heidrun (Germany)  
Shamir, Ariel (Israel)  
Skala, Vaclav (Czech Republic)  
Sochor, Jiri (Czech Republic)  
Teschner, Matthias (Germany)  
Theoharis, Theoharis (Greece)  
Veltkamp, Remco (The Netherlands)  
Weiskopf, Daniel (Canada)  
Wu, Shin-Ting (Brazil)  
Wyvill, Brian (Canada)  
Zemcik, Pavel (Czech Republic)



## WSCG 2007 Board of Reviewers

Adzhiev, Valery (U.K.)  
Alessandro, Piva (Italy)  
Ammon, Lorenz (Switzerland)  
Ancuti, Cosmin (Belgium)  
Andreadis, Ioannis (Greece)  
Andujar, Carlos (Spain)  
Aspragathos, Nikos (Greece)  
Avenueau, Lilian (France)  
Bamidis, Panagiotis (Greece)  
Bargouli, Maria (Greece)  
Bartz, Dirk (Germany)  
Battiato, Sebastiano (Italy)  
Bekaert, Philippe (Belgium)  
Bellon, Olga (Brazil)  
Benes, Bedrich (USA)  
Bengtsson, Ewert (Sweden)  
Beyer, Johanna (Austria)  
Bieri, Hanspeter (Switzerland)  
Bilbao, Javier J. (Spain)  
Biri, Venceslas (France)  
Bottino, Andrea (Italy)  
Bouatouch, Kadi (France)  
Bourdin, Jean-Jacques (France)  
Brunet, Pere (Spain)  
Bruni, Vittoria (Italy)  
Brunnet, Guido (Germany)  
Buehler, Katja (Austria)  
Butz, Andreas (Germany)  
Callieri, Marco (Italy)  
Clasen, Malte (Germany)  
Coleman, Sonya (U.K.)  
Coquillart, Sabine (France)  
Crosnier, Andre (France)  
Daniel, Marc (France)  
Davis, Larry (USA)  
de Geus, Klaus (Brazil)  
Deussen, Oliver (Germany)  
Dingliana, John (Ireland)  
du Buf, Hans (Portugal)  
Duce, David (U.K.)  
Durikovic, Roman (Slovakia)  
Egges, Arjan (The The Netherlands)  
Eisemann, Martin (Germany)  
Erbacher, Robert (USA)  
Faudot, Dominique (France)  
Feito, Francisco (Spain)  
Felkel, Petr (Czech Republic)  
Ferguson, Stuart (U.K.)  
Ferko, Andrej (Slovakia)  
Fernandes, Antonio (Portugal)  
Fischer, Jan (Germany)  
Flaquer, Juan (Spain)  
Galin, Eric (France)  
Galo, Mauricio (Brazil)  
Ganovelli, Fabio (Italy)  
Garcia-Alonso, Alejandro (Spain)  
Gautron, Pascal (France)  
Giannini, Franca (Italy)  
Goebel, Martin (Germany)  
Gonzalez, Pascual (Spain)  
Grammalidis, Nicolaos (Greece)  
Gresh, Donna (USA)  
Groeller, Eduard (Austria)  
Gutierrez, Diego (Spain)  
Habbecke, Martin (Germany)  
Haber, Tom (Belgium)  
Hadwiger, Markus (Austria)  
Harding, Bruce (USA)  
Haro, Antonio (USA)  
Hast, Anders (Sweden)  
Hauser, Helwig (Austria)  
Havran, Vlastimil (Czech Republic)  
He, Xiangjian (Australia)  
Hege, Hans-Christian (Germany)  
Horain, Patrick (France)  
Hornung, Alexander (Germany)  
House, Donald (USA)  
Chen, Min (U.K.)  
Chin, Seongah (Korea)  
Chover, Miguel (Spain)  
Chrysanthou, Yiorgos (Cyprus)  
Iglesias, Andres (Spain)  
Ihrke, Ivo (Germany)  
Jaillet, Fabrice (France)  
Jansen, Frederik (The The Netherlands)  
Joan-Arinyo, Robert (Spain)  
Kähler, Ralf (Germany)  
Kalantari, Leila (Canada)  
Karabassi, Evaggelia-Aggeliki (Greece)  
Kersten, Marta (Germany)  
Kinuwaki, Shinichi (Germany)  
Kjelldahl, Lars (Sweden)  
Klajnsek, Gregor (Slovenia)  
Klosowski, James T. (USA)  
Knight, Michael (U.K.)  
Kobbelt, Leif (Germany)  
Kohlmann, Peter (Austria)  
Kolcun, Alexej (Czech Republic)  
Kruijff, Ernst (Austria)  
Lamecker, Hans (Germany)  
Lanquetin, Sandrine (France)  
Larboulette, Caroline (Austria)  
Lee, Seungyong (Korea)  
Leitao, Miguel (Portugal)  
Lensch, Heindrik (Germany)  
Lewis, J.P. (USA)  
Linz, Christian (Germany)  
Magillo, Paola (Italy)  
Magnor, Marcus (Germany)  
Malik, Muddassir (Austria)  
Mandl, Thomas (Germany)  
Mantler, Stephan (Austria)  
Matey, Luis (Spain)  
Mattausch, Oliver (Austria)  
McMenemy, Karen (U.K.)  
Megali, Giuseppe (Italy)  
Meneveaux, Daniel (France)  
Moccozet, Laurent (Switzerland)  
Mokhtari, Marielle (Canada)  
Mollá Vayá, Ramón (Spain)  
Montrucchio, Bartolomeo (Italy)  
Mould, David (Canada)  
Mudur, Sudhir (Canada)  
Muller, Heinrich (Germany)  
Muntean, Adrian (Germany)  
Myszkowski, Karol (Germany)

Nedel, Luciana P. (Brazil)  
Neveu, Marc (France)  
Nielsen, Frank (Japan)  
Ogayar, Carlos,J. (Spain)  
Oliveira, Manuel (U.K.)  
Pasko, Alexander (Japan)  
Patel, Daniel (Austria)  
Patow, Gustavo (Spain)  
Paulin, Mathias (France)  
Pavic, Darko (Germany)  
Pedrini, Helio (Brazil)  
Peroche, Bernard (France)  
Pettifer, Steve (U.K.)  
Platis, Nikos (Greece)  
Plemenos, Dimitri (France)  
Podorelec, David (Slovenia)  
Ponchio, Federico (Italy)  
Popov, Stefan (Germany)  
Prakash, Edmond (U.K.)  
Pratikakis, Ioannis (Greece)  
Přikryl, Jan (Czech Republic)  
Puppo, Enrico (Italy)  
Purgathofer, Werner (Austria)  
Rauterberg, Matthias (The The Netherlands)  
Redon, Stephane (France)  
Remolar, Inmaculada (Spain)  
Renaud, Christophe (France)  
Revelles, Jorge (Spain)  
Ribelles, Jose (Spain)  
Robert, Philippe (Switzerland)  
Rodeiro, Javier (Spain)  
Rodrigues, Marcos (U.K.)  
Rojas-Sola, José Ignacio (Spain)  
Rokita, Przemyslaw (Poland)  
Rossignac, Jarek (USA)  
Roth, Peter,M. (Austria)  
Rudomin, Isaac (Mexico)  
Sanna, Andrea (Italy)  
Sbert, Mateu (Spain)  
Scateni, Riccardo (Italy)  
Segura, Rafael (Spain)  
Sellent, Anita (Germany)  
Semwal, Sudhanshu (USA)  
Shamir, Ariel (Israel)  
Scheiblauer, Claus (Austria)  
Schilling, Andreas (Germany)  
Schmidt, Thomas (Germany)  
Schneider, Bengt-Olaf (USA)  
Schumann, Heidrun (Germany)  
Sips, Mike (USA)  
Sirakov, Nikolay Metodiev (USA)  
Sochor, Jiri (Czech Republic)  
Solis, Ana Luisa (Mexico)  
Sousa, A.Augusto (Portugal)  
Stroud, Ian (Switzerland)  
Suarez Rivero, Jose (Spain)  
Suescun, Angel (Spain)  
Szekely, Gabor (Switzerland)  
Štulić, Radovan (Serbia and Montenegro)  
Tang, Wen (U.K.)  
Tawara, Takehiro (Japan)  
Tecchia, Franco (Italy)  
Teschner, Matthias (Germany)  
Theußl, Thomas (Austria)  
Tobler, Robert (Austria)  
Tonet, Oliver (Italy)  
Turini, Giuseppe (Italy)  
Tytkowski, Krzysztof (Poland)  
Vanecek, Petr (Czech Republic)  
Vasa, Libor (Czech Republic)  
Veiga, Luis (Portugal)  
Veltkamp, Remco (The The Netherlands)  
Vergeest, Joris (The The Netherlands)  
Vitulano, Domenico (Italy)  
Wan, Taoruan (U.K.)  
Weigel, Christian (Germany)  
Weiskopf, Daniel (Canada)  
Wenger, Thomas (Switzerland)  
Wu, Shin-Ting (Brazil)  
Wyvill, Brian (Canada)  
Zach, Christopher (Austria)  
Zachmann, Gabriel (Germany)  
Zalik, Borut (Slovenia)  
Zemcik, Pavel (Czech Republic)  
Zhu, Ying (USA)  
Zimeras, Stelios (Greece)

# Contents

## Part I

Paper Title Code	Page
B05 Birkholz,H.: Patch-Trees for Fast Level-of-Detail Synthesis (Germany)	1
D19 Garcia,R., Urena,C., Lastra,M., Montes,R., Revelles,J.: A study of Incremental Update of Global Illumination Algorithms (Spain)	7
E03 Nakamura,N., Inoue,Y., Teshima,Y., Nishio,K., Kobori,K.: A Study on Generation Method from Boundary Representation Model to Binary Voxel Model by Using GPU (Japan)	15
E59 Zotti,G., Wilkie,A., Purgathofer,W.: A Critical Review of the Preetham Skylight Model (Austria)	23
E73 Domonkos,B., Egri,A., Foris,T., Juhasz,T., Szirmay-Kalos,L.: Isosurface Ray-casting for Autostereoscopic Displays (Hungary)	31
F37 Gumbau,J., Ripolles,O., Chover,M.: LODManager: A Framework for Rendering Multiresolution Models in Real-time Applications (Spain)	39
H19 Toth,B., Szirmay-Kalos,L.: Fast Filtering and Tone Mapping using Importance sampling (Hungary)	47
D97 Ko,P.K., Lee,J.-J., Choi,N.-S., Lee,B.-G.: Generalization of (2n+4)-point Approximating Subdivision Scheme (Korea)	53
A29 Eisemann,M., Grosch,T., Magnor,M., Müller,M.: Automatic Creation of Object Hierarchies for Ray Tracing Dynamic Scenes (Germany)	57
B79 Veleba,D., Felkel,P.: Detection and correction of errors in surface representation (Czech Republic)	65
D59 Sakalauskas,T.: Silhouette Partitioning for Height Field Ray Tracing (Lithuania)	73
E19 Ungvichian,V., Kanongchaiyos,P.: Finding Thin Points in an Abstract Cellular Complex (Thailand)	81
E41 Ungvichian,V., Kanongchaiyos,P.: 3-D Object Extraction Using Volume Computation (Thailand)	89
F03 Lai,G., Christensen,N.J.: A Compression Method for Spectral Photon Mapping (Denmark)	95
G53 Loke,R.E., Jansen,F.W.: Maintaining Sharp Features in Surface Construction for Volumetric Objects (Netherlands)	103
A13 Ausheva,N., Demchyshyn,A.: Fractal Modeling of Vacuum Arc Cathode Spots (Ukraine)	111
A03 Qian, X.,Zhao,Z.,Thorn,R.: Rapid Development of Virtual Environments: A systematic approach for interactive design of 3D graphics (United Kingdom)	117
A37 Huebner,K.: Object Description and Decomposition by Symmetry Hierarchies (Germany)	125

# Contents

## Part II

Paper Title Code	Page
A71 Pospisilova,R.: Occlusion Detection and Surface Completion in 3D Reconstruction of Man-made Environments (Czech Republic)	133
C17 Oba,S., Ikai,T., Aoki,S., Yamashita,T., Izumi,M., Fukunaga,K.: Viewpoint Selection Based on Fechner Type Information Quantities for 3D Objects (Japan)	141
C61 Krajíček,V., Pelikán,J., Horák,M.: Measuring and Segmentation in CT Data using Deformable Models (Czech Republic)	149
D71 Oh,S., Yeo,W.: Comparative Navigation System for Collaborative Project (Japan)	153
D79 Khanduja,G., Bijaya,B.K.: Using Graphics Hardware for Multiple Datasets Visualization (United States)	161
F41 Hashimoto,M., Miyamoto,K.: Physically Realistic Interface for a User Inside VR (Japan)	169
A23 Nassiri,N., Powell,N., Moore,D.: Virtual Environments and Human-Computer Interaction Anxiety: An Experimental Study (United Arab Emirates)	177
F13 Gonzales,M.R.M., Wu,Sh.-T.: Using Simplified Meshes for the Crude Registration of Two Partially Overlapping Range Images (Brazil)	183
F89 Thanasootornlerk,K., Kanongchaiyos,P.: An Image Matching Using Critical-Point Filter and Level Set Analysis (Thailand)	191
G17 Hussein,R., Mckenzie,F.: Automatic Identification of ambiguous Prostate Capsule Boundary Lines using Shape Information and Least Squares Curve Fitting Technique (United States)	199
G41 Popov,A.T.: Fuzzy Mathematical Morphology and Its Applications to Colour Image Processing (Bulgaria)	205
H47 Martinez,D., Hernandez,J.T., Florez,L.: Contributions to Colon Segmentation Without Previous Preparation in Computer Tomography Images (Colombia)	211
D37 Rodriguez,J.,Linares,P.,Reigosa,A.,Laya,D.,Urra,E., Saldivia,F.: Estimating the Tumor-breast Volume Ratio from Mammograms (Venezuela)	219
F23 Sharma,O., Mioc,D., Anton,A.: Feature Extraction and Simplification from Colour Images based on Colour Image Segmentation and Skeletonization using the Quad-Edge Data Structure (Denmark)	225
G89 Vergeest,J.S.M., Song,Y., Kroes,T.: Associating 6 DoF Sensor Data to 3D Scan View Registration (Netherlands)	233
H37 Wang,Y., Turner,M.J., Perrin,J., Hewitt,W.T.: Evaluation of Different Vibration Visualization Modes for Line Tracking (United Kingdom)	241
H67 Tantisiriwat,W., Sumleeon,A., Kanongchaiyos,P.: A Crowd Simulation Using Individual-Knowledge-Merge based Path Construction and Smoothed Particle Hydrodynamics (Thailand)	249
H71 Pakdee,S., Kanongchaiyos,P.: Biped Cartoon Retrieval Using LBG-Algorithm Based State Vector Quantization (Thailand)	257

# Occlusion detection and surface completion in 3D reconstruction of man-made environments

Radka Pospíšilová  
Masaryk University, Faculty of  
Informatics  
Botanická 68a  
602 00, Brno, Czech Republic  
xpospis5@fi.muni.cz

## ABSTRACT

This paper presents the methods for the quality improvement of the man-made environments reconstruction. The 3D model is created from the set of images without any user interaction. Some generic assumptions about the scene geometry are used; for example continuity of surfaces, piecewise planar scene or frequent parallelism and orthogonality between planes and lines. This information is employed to reduce the ambiguity of the 3D reconstruction and for the occlusion recovery. Method for the completion of the occluded surfaces is presented. First, the position of missing parts of the scene is detected and then the gaps are filled by the most suitable shape.

## Keywords

3D reconstruction, occlusion, surface completion.

## 1. INTRODUCTION

In this paper we focus on the quality improvement of the 3D reconstruction of man-made environments. The 3D model of the scene is created without any user interaction from the set of images. We investigate the possibilities of the completion ambiguous and unreconstructed surfaces. Concretely, we focus on full automatic completion of the gaps in 3D model, which was caused by occlusion. An occlusion is a covering of far surfaces by the closer ones. If the covered parts of the scene are visible in simply one image, we have only the information about texture and shape of their projection. If the covered parts are not visible on any photography, the surface is completely unknown.

The occlusion causes problems in most systems for scene reconstruction. It makes the reconstruction

process more erroneous and it creates holes in the 3D models, so the result of the reconstruction is visually unsatisfactory. Small holes in the surface are usually filled by interpolation, but if the object is indented, the interpolation gives very bad results. If the occlusion occurs in reconstruction of unknown scene, it is impossible to fill occluded surfaces because the geometry of the scene is unknown.

We focused on situation where we have some assumptions or information about the scene, so it is possible to use them. Man-made environments often satisfy some simple and useful assumptions and it is relatively easy to employ them in the reconstruction. In our system we used algorithm for 3D reconstruction which is able to reconstruct relatively exactly the neighbourhood of the gaps and then the best shape for the occluded surface is estimated.

Nowadays we develop the system for the reconstruction of piecewise planar scenes with straight lines. We integrated the algorithm of filling holes into the reconstruction process, but it is possible to use it for formerly completed models too.

## 2. RELATED WORK

Most of the existing systems for 3D reconstruction don't compute the shape of the occluded parts in the scene. Usually if the occlusion is detected, the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.  
Copyright UNION Agency – Science Press, Plzen, Czech Republic.

missing surface is completed by the interpolation. So the 3D points on occluded surface are computed from border points only. Most existing sophisticated approaches detect and solve the occlusion in range image from 3D scanner, for example [Cas02a] or [Del01a].

### Occlusion Detection in Range Image

Some approaches only detect the occlusion and then compute the next best position of the range sensor [San99a]. The found position of the sensor is optimal for the holes completion.

The other methods also detect the occlusion and then they can complete some types of occluded surfaces. In [Del01a] the assumption on continuous surfaces is used and algorithm is focused on indoor planar scene. If some planar surface in the 3D model is splitted to the smaller parts by occlusion, then the optimal position of the plane is computed from these small parts. Then the parts are connected so the plane lies over the occluded gaps. Partly occluded non-planar surfaces (for example parts of cylinder or sphere) can be reconstructed too. When the shape and type of geometric object is recognized from the visible parts, the surface can be extended over the occluded parts.

The approach of Castellani et al. [Cas02a] is specialized on reconstruction of the planes, corners and edges of partially occluded simple-shape objects. Some types of larger occlusion can be solved too, for example object composed of one or more polygons and their parts. In Castellani's algorithm first the range image is segmented and the planes are divided into the background and foreground. Then the end-points of the edges between the foreground and the background are found and finally the surface and edges are reconstructed.

### Occlusion and 3D Reconstruction from Images

If we reconstruct 3D model from the set of images, the occlusion can cause many problems. Especially many false 2D points are produced by edge detector. When the correspondences between the points are searched, all correspondences which contain these false 2D points are obviously incorrect, so the algorithm for the structure and motion estimation must be really robust due to a big amount of outliers. Another problem is a detection of the occlusion and finding of the borders of the obscured parts. For these two reasons there aren't many approaches focused on computing the occluded surfaces from images.

Approaches with **model fitting** are the exceptions. We have a set of predefined models, where some

properties of the models are known; for example the shape, color or texture. If we fit the models into the scene, it is possible to compute the small invisible parts of the surface. Model fitting is very useful especially in the reconstruction of an architecture object, because it is possible to complete the occluded parts of the columns, windows, doors, etc.

Another advantage of model fitting approach is more precise reconstruction of repeated models in the scene. For example buildings often contain several windows with the same shape, so it is possible to choose the best displayed windows and replace another distorted and occluded models by its. More information can be found for example in [Dic01a] or [Wer02b].

The occlusion must be detected also in the **dense depth estimation**. First, pictures are rectified so the epipolar lines are aligned horizontally. The corresponding points for each pixel are searched along the epipolar line. If the uniqueness of the correspondences is corrupted (one point has more then one corresponding point), then the occlusion probably occurs and space between these two points was displayed only in one image.

### 3. ASSUMPTIONS ON THE SCENE

The application of the assumptions on the scene geometry is very common in reconstruction of the man-made environments. Various assumptions and prior information can be used for calibration of cameras and for the quality improvement of the 3D model. Such prior information does not only solve the projective ambiguity in the reconstruction but also usually stabilizes the sensitive reconstruction process.

The reconstruction of architecture objects without any information is relatively complicated task. Searching for the point correspondences is often unreliable for some areas in the images. It is caused especially by the homogeneous region in the scene, repeated pattern, various lighting condition and occlusion. If we have only few images of the scene, then the problem is getting more serious.

#### Assumptions about the Architecture

Only simple and easy applicable geometric constraints we used for the reconstruction of the scene with the occluded parts. There is a list of employed assumptions:

- **Large coplanar regions** – man-made scenes usually contains coplanar regions. Large regions can be discontinued by wedge or protruded objects. These objects can be already coplanar.

- **Continuity of surface** – each plane has a boundary which is composed of the lines. If the occlusion doesn't occur then each border line has two planes which go through the line. If any border line lies only on one plane, probably the second plane isn't reconstructed, so the best position and orientation of the plane must be found.
- **Occluded surface doesn't change shape abruptly** – if this assumption is corrupted, the holes can't be reconstructed correctly.
- **Perpendicularity** of planes and lines.
- **Parallelism** of planes and lines.
- **Straight lines** – the scenes of this type usually contain big amount of straight lines. So we can assume that slightly curved lines are really straight and curvature is caused by the inaccuracy or distortion of the cameras.
- **Symmetry** – the symmetry along some axis frequently occurs in architecture. Usually it is suitable to search symmetry along the axis, which is parallel to one of the dominant direction in the scene. The symmetry can be applied even in the situations where the other assumptions are useless because of a large area of the occluded surface. Unfortunately it is hard to detect the symmetry if the object is too complicated.



**Figure 1. Example of architecture object (Kroměříž Chateau), which fulfils these assumptions.**

#### **4. 3D RECONSTRUCTION WITH GAPS COMPLETION**

Our algorithm for reconstruction of the scene is inspired by the some methods which are described for example by Tomáš Werner [Wer02a], Baillard and Zisserman [Bai99a] etc. We extend and modify these existing approaches, so it is possible to fill the holes and ambiguous parts of the 3D model.

##### **Points and Lines Reconstruction**

First, feature points and lines are matched among the images. We reconstruct both lines and points because lines are not corrupted by the occlusion so much as

the points. On the other hand three views are necessary for the reconstruction of lines and reconstruction of the points needs only two views. Therefore we can obtain the positions of 3D points from those parts of the scene where the line reconstruction is impossible.

Trifocal tensor is used for 3D reconstruction of lines and for computation of camera matrix. It is estimated using the standard robust methods (see [Har03a] for full details). Now the camera matrices and a set of 3D points and lines must be transformed to metric reconstruction.

##### *4.1.1. Upgrade to metric reconstruction*

Our system must be able to make quality 3D model from relatively small amount of images; therefore the geometric constraints are employed in camera calibration and metric rectification process. Man-made environments usually contain three dominant directions in the scene and these directions are perpendicular. First, the principal directions (represented by points at infinity) are computed from their images – vanishing points. The plane at infinity is identified and the projective reconstruction is updated to affine. Then the model is transformed so that the principal directions are perpendicular. Finally, the scaling in each principal direction is computed using assumption on the square pixels.

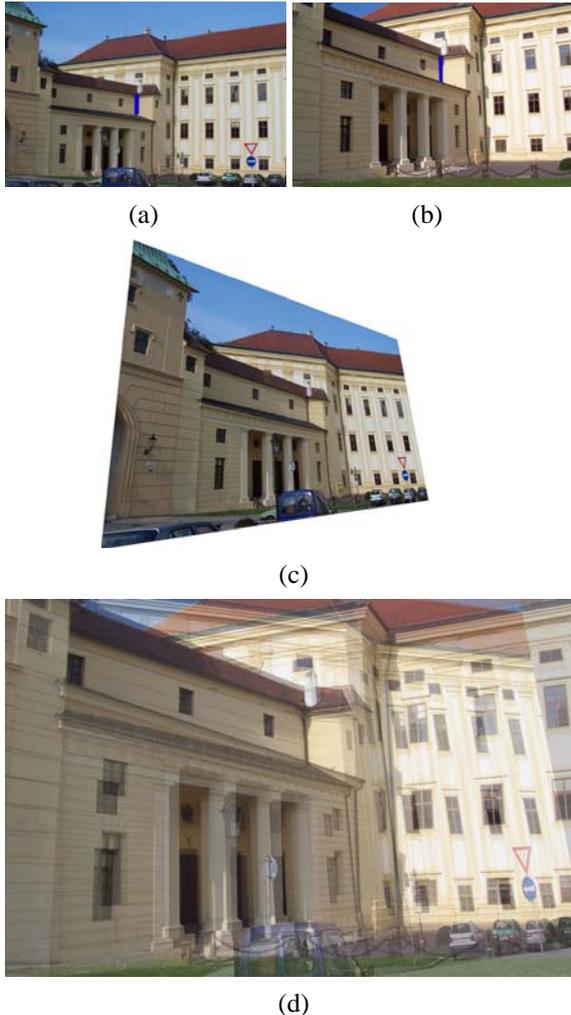
##### **Plane and Polygon Fitting**

Some approaches detect planes and polygons directly in the cloud of 3D points, which are reconstructed using dense depth estimation. However these methods are inaccurate for 3D points, which are situated near occluded parts of the scene. Therefore we use the different plane fitting algorithm which uses plane-sweep method.

In our approach we have used two methods for plane fitting, it is plane-sweep algorithm and RANSAC [Fis81a]. Details about plane-sweep method are described in [Bai99a]. We use the reconstructed feature lines and points for the detection of planes. Big advantage of the plane-sweep method is an ability to determine the scene plane even in the absence of any reconstructed points of that plane. On the other hand if two views of the scene are too different, the method can fail. It is usually caused by significant movement and rotation of the camera or by reflection. Second problem is detection of homogeneous planes without texture because comparison of the points on such planes is very inaccurate. Therefore it is necessary to verify the reconstructed planes and to combine plane-sweep method with RANSAC.

#### 4.2.1. Plane-sweep

In several next sections we introduce the plane-sweep method and we show the extension of this method by searching the boundary of the plane. First the position of the plane is searched along each reconstructed 3D line. All possible positions along each line are explored and optimal angle for position of plane is chosen. If the line at infinity is used for searching, then all parallel planes in given direction are searched.



**Figure 2. Plane-sweep applied along the line which is marked by blue color in the image (a) and (b). Picture (c) was created by application of homography on image (a). This homography correspond to real position of the plane on the left of the blue line. Image (d) is obtained as a picture (b) which is superimposed by the picture (c). It can be seen that only this plane is "focussed".**

The position of the new plane is detected by similarity measure between first image and second image, which is transformed by homography. Homography is point-to-point mapping between the images and it is induced by the real 3D position of the plane. So the homography determines the correspondence between the points and their local neighbourhood is compared and correlation of the points is computed. The sum of the correlation scores is compared for all possible directions and plane is placed to that position, where the similarity score function has the maximum. We use the same similarity function as a Baillard in [Bai99a] and we use only key points for similarity computation too. These key points are obtained by applying an edge detector with very low threshold on the gradient.

#### Boundary Detection

One of the most serious problems in plane-sweep method is a lack of reconstructed 3D lines. It is necessary to identify the whole boundary for each plane and it means detection of all visible lines and the estimation of some occluded lines in the scene.

Usually only about 30-50 % of the important feature lines is found for average intended scene during the lines reconstruction. Edge detector can't find all borders, because they can be blurred or obscure. Moreover some borders are occluded on some photos so the reconstruction of the line can't be done. We show the method for detection of these missing lines.

#### 4.3.1. Searching for new lines by planes intersections

If the line can't be reconstructed by the trifocal tensor, the simplest and most precise method is to find it as an intersection of two planes. This method was first used in [Bai99a].

#### 4.3.2. Creation of boundary from similarity values

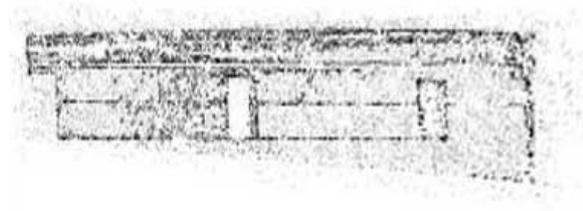
If occlusion didn't occur and all planes are sharp, then it is possible to find all missing lines as the plane intersection. However occlusion and presence of blurred planes makes the detection of lines more complicated. In this case the only possibility is to obtain the line from correlation values. It is obvious that this procedure produces lines with lower accuracy.

The plane-sweep method is very sensitive to displacement of the lines, and the lower accuracy of line can cause failure of the next plane detection. However it is not significant problem, because lines detected only from correlation values lie on planes, where the next plane is occluded or unreconstructable. So the plane can't be found even along the accurate line.

#### 4.3.2.1 Computation of rough border

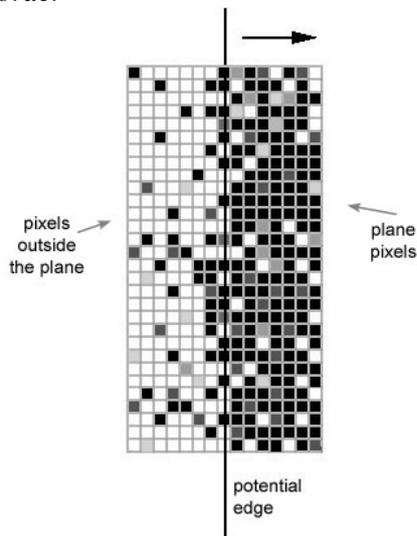
When we want to estimate the boundary of plane, it is necessary to include correlation scores of all pixels (or regular samples of pixels) in computation. For each pixel the weight values are computed from the gradient of the pixel and it defines its significancy. The key points have the weight equal to 1 and for the pixels in homogeneous region the weight tends to zero.

Then the image of correlation values is segmented into the parts of plane and remaining parts. Before the segmentation it is necessary to smooth the correlation values. The thresholding with iterative threshold selection is used for segmentation and then rough borders between regions are computed. If the closed boundary inside the plane is found, it is excluded out of the plane.



**Figure 3. Correlation values for identified plane are represented by pixel darkness. The magnitude of the similarity scores is modified by the gradient of the pixel.**

#### 4.3.2.2 Computation of exact position of the border



**Figure 4. Searching for the optimal edge, white points are pixels with zero correlation; black points are pixel with maximal correlation.**

Found borders are usually too thick so it is necessary to compute their exact position. If the real line (reconstructed by trifocal tensor) exists in this thick edge, it is set as a border of the actually reconstructed plane. If such line isn't available, its position must be computed from the correlation values only. The thick border is divided into straight segments and in each segment the optimal position of the border is computed. We choose the outline which divides the local neighborhood in the best way into a region densely covered by points with high correlation and a region with low correlation values. Lines are searched in the direction parallel to the actually tested border segment.

Function  $C_E$  is maximized over all position of lines:

$$C_E = \left( \sum_{\text{Cor}(P_i) \geq T} W(\text{Cor}(P_i)) + \sum_{\text{Cor}(O_i) < T} W(T - \text{Cor}(O_i)) \right) - \left( \sum_{\text{Cor}(P_i) < T} W(T - \text{Cor}(P_i)) + F \sum_{\text{Cor}(O_i) \geq T} W(\text{Cor}(O_i)) \right),$$

where  $P_i$  are pixels on plane,  $O_i$  are pixels outside the plane,  $\text{Cor}(X)$  is the correlation value of the point,  $T$  is the threshold, which divides pixels to high correlated and low correlated and  $W$  is weight of pixels, which depends on the gradient. Because minimizing of the outliers has a higher precedence than maximizing the high correlated plane pixels, the weighting coefficient  $F$  is used to tune the algorithm; typical values are  $F \in [1, 2]$ .

Problems can arise if the actually reconstructed plane forms angle near  $180^\circ$  with the adjacent plane or if no keypoints are present near the border. In this situation the boundary of region can be detected wrongly.

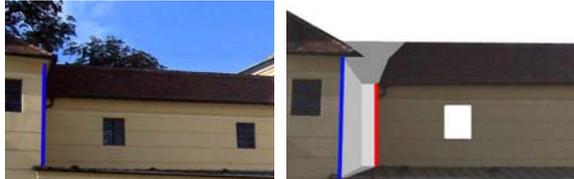
#### Verification of the line position

The position of each border line is verified by its re-projecting to the images. The level of gradient is explored and the border line is accepted or rejected. Simultaneously the edges are divided into several types (see section 4.5.1) because the correct classification of edges is essential for the reconstruction of the occluded parts.

If the level of gradient for some edge is low in all images, it is possible to search the edge in the next  $N$  suboptimal solutions of cost function  $C_E$ . If no other suitable border is found, the first detected line is used.

## Occlusion Detection

During the reconstruction of the planes we often meet the situation where some border line belongs only to one plane or some lines are not connected to other lines. It can be caused by various reasons; the edge detector doesn't find the border line, the occlusion appears or the plane fitting algorithm fails. Another eventuality can be presence of non-planar object in the scene. For the completion of „invisible“ parts it is necessary to classify all edges to several groups.



(a) (b)

**Figure 5. Picture (a) is view with the occlusion; in picture (b) the occluded parts are completed. The blue colour denotes self-occluded edge and red is false edge. These two lines have same image in picture (a). The red edge is computed from two photos during boundary estimation.**

### 4.5.1. Edge classification

We can distinguish three main types of edges:

**Normal edge** – It creates boundary for more than one plane, so the occlusion doesn't occur in the neighbourhood of this edge.

**Self-occluded edge** – If one plane occludes the other plane and these two planes are connected by common edge, we denote this edge as **self-occluded**. We can easily recognize this type of edge because during reconstruction it belongs only to one plane and when the edge is verified, it is displayed in more than one image. For estimation of the plane behind the self-occluded edge the assumption on the continuous shape can't be used, because the shape of the surface is usually changed abruptly on this edge.

**False edge** – We can recognize the unreconstructed parts of the plane by the presence of the **false edges**. False edges don't really exist in the scene; their images in photos are caused by the occlusion. All false edges are computed only from correlation values. The edge is classified as false if it has image in at most one photo. The false edges can't have same position in more than one photo because the location of the edge depends on direction of camera, so it is necessary to exclude too similar views from edge classification.

If the occlusion was caused by other parts of the scene, then the image of false edge is coincident with some self-occluded edge in just one photo. If false edge hasn't corresponding self-occluded edge then its presence could be caused by:

- **Covering by unreconstructed objects** – for example the buildings can be covered by branches of a tree. This results in the holes in the scene which are bounded by irregular false edges without corresponding self-occluded edge.
- **Errors in reconstruction** – some parts of the planes are not reconstructed, because the plane detection algorithm failed.

If we can't reconstruct the object in the foreground, evidently it is impossible to distinguish two last situations – covering by unreconstructed object and errors in the reconstruction.

## Algorithm for Reconstruction of Piecewise Planar Scene

First, the empty list  $\mathbf{P}$  of planes in the scene is created. Each plane contains reference to the lines, which create the boundary of the plane. Edges are stored in four lists:

- $\mathbf{E}_{\text{single}}$  contains normal edges, which don't belong to any plane.
- $\mathbf{E}_{\text{normal}}$  contains edges, which are assigned to more than one plane.
- $\mathbf{E}_{\text{false}}$  contains false edges.
- $\mathbf{E}_{\text{self-occ}}$  contains self-occluded edges.

### 4.6.1. Algorithm

1. Metric 3D reconstruction of the points and lines is computed.
2. The list of all reconstructed lines  $\mathbf{E}_{\text{single}}$  is created, lines are sorted descending by length and the lines at infinity are inserted in the first positions of the list.
3. The first edge from  $\mathbf{E}_{\text{single}}$  list is taken and the line is searched along this edge (section 4).
  - If there isn't any plane, the edge is removed from  $\mathbf{E}_{\text{single}}$  list and if the line isn't at infinity, it is moved to the  $\mathbf{E}_{\text{self-occ}}$  list. New plane along the next edge is searched.
  - If the plane is found, it is added to  $\mathbf{P}$  list. If number of plane for actual edge is equal to 2 and if line isn't at infinity, then the edge is added to the  $\mathbf{E}_{\text{normal}}$  and removed from  $\mathbf{E}_{\text{single}}$  list.
4. The approximate positions of the edges are found (section 4.3.2.1). If some existing lines

have suitable positions then they are chosen as the border lines and the shape of boundary and number of plane for the edge is corrected.

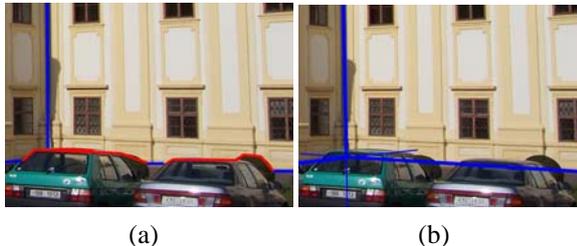
5. If the boundary of the plane isn't closed, the intersection with other plane is searched. If new line is found, it is added to  $E_{\text{single}}$  list and the boundary of these planes is corrected by this edge. If the  $E_{\text{single}}$  list isn't empty, go to 3.
6. If  $E_{\text{single}}$  list is empty, missing borders are computed for each plane in  $P$ , which doesn't have closed boundary. Position of these borders is computed from the correlation values (sections 4.3.2.1, 4.3.2.2). Again the next planes are searched along each new border line. The type of the line is recognized (section 4.5.1).
7. If no more edges can be found, then the reconstruction process is finished. Now the occluded and unreconstructed surfaces must be filled.

## 5. OCCLUSION RECOVERY

The way of completing covered parts is different for the object which is occluded by the other object or by itself. Further it depends on the shape of the neighbourhood of holes.

If the occluded parts are too complex, it is often impossible to recovery the shape. But it's not surprising because only very little information is available about the occluded part.

### Searching for Correct Boundary



**Figure 6. In picture (a) the planes have boundary from normal (blue) and false (red) edges. In picture (b) the false edges are removed and normal edges are extended over the gap and corners are found.**

The most common situation is incomplete boundary of plane, where some border edges are false or missing. If the border lines are false, then the real existing lines must be extended over the gap. Sometimes new border line must be added using geometric assumptions about the scene. Finally the false edges are removed from the boundary.

### Searching for New Planes

If we have two self-occluded edges with common point, then these two edges determine the position of the occluded plane.

Completion of the plane is more difficult if we have only one self-occluded edge for identification of the plane position. In this situation the coordinates of the plane must be determined only from geometric assumptions (usually the perpendicularity, symmetry or parallelism). Remaining border lines of new plane are computed by intersection with other planes or again by geometric assumptions.

### Searching for New Planes without any reconstructed line

If some small wedge or protruded objects appear on the larger plane (for example the window on the wall), then it is usually imposible to reconstruct any line of this object. This situation is recognized as a gap in the plane, which is bounded by normal and self-occluded edges (wedge objects) or normal and false edges (protruded objects). Without at least one reconstructed 3D line of new plane we can't use the plane-sweep method for plane detection, so we only use the geometric assumptions and the new plane is searched as a parallel in the set of reconstructed 3D points. Other small planes, which connect the large plane and its gap, are perpendicular to them.

### Algorithm for occlusion recovery

1. For each plane all identified edges (both inner and outer border lines) are revised:
  - The sequence of edges is checked.
  - If the sequence is corrupted or if boundary contains false edges, then real edges must be extended and false edges are removed.
  - If the closed boundary is created only by false edge, it can be removed from the reconstruction.
2. New planes along the self-occluded edges are searched and for each new plane existence of intersections with other planes is tested. If the intersection is found, then the new edge is added.
3. For planes with the gaps the best shape is estimated for wedge and protruded object.
4. All planes without closed outer boundary are removed from the reconstruction.
5. If the occluded object was displayed in one image, then the texture can be extracted. If this part of the scene doesn't have an image, the surface is textured by blurred texture of the neighbourhood of the gap.

## 6. CONCLUSIONS

We have presented a method for occlusion detection and recovery in reconstruction of piecewise planar scene. The method is based on fitting planes in the cloud of reconstructed points and lines. If there isn't enough information for fitting the plane in common way, then the type of occlusion is estimated and the most suitable shape of the plane is determined according to neighbourhood of the plane and assumptions on the scene.

The big advantage of this method is ability to estimate occluded surfaces in full automated reconstruction without any high-level information about the scene. However, there are still limitations and some parts of method must be improved. Indeed, the method strongly depends on border detection and occlusion classification, but these parts of method can be unreliable due to lighting condition, weak contrast etc. We still work on improvement of these parts of algorithm, but the problems with bad light condition, contrast, shadows or reflections are common in all computer vision applications and they can't be solved definitively.

## 7. FUTURE WORK

We have focused only on the 3D reconstruction of piecewise planar scene, however various non-planar objects occur frequently in the man-made environments. If such an object is in the scene, the reconstruction will be incorrect. So we plan to extend this approach by reconstruction of arcs and some curved surfaces.

Further, system of geometric constraints can be extended by new, user defined information. These constraints could be added before the reconstruction or interactively in the process of reconstruction, when the algorithm requires additional information for ambiguous surfaces.

Algorithm for detection of plane shape is reliable for textured planes, but if the texture is missing then the borders are inaccurate. This problem can be partially solved by the extension of algorithm by texture and color analysis. Results of this analysis can then be used for chaining of the planes and polygons, for

detection of the objects outside the plane and for the better estimations of the texture of occluded surfaces.

## 8. ACKNOWLEDGMENTS

This work was supported by The Ministry of education of The Czech Republic, Contract No. LC06008 and by The Grant Agency of The Czech Republic, Contract No. 201/07/0927.

## 9. REFERENCES

- [Bai99a] Baillard, C. and Zisserman, A. Automatic reconstruction of piecewise planar models from multiple views. Conference on Computer Vision and Pattern Recognition, pp. 559-565, June 1999.
- [Cas02a] Castellani, U., Livatino, S., and Fisher, R. B. Improving environment modelling by edge occlusion surface completion. Int. Symp. on 3D data processing, visualization and transmission (3DPVT), Padova, Italy, pp. 672-675, 2002.
- [Del01a] Dell'Acqua, F. and Fisher, R.B: Reconstruction of surfaces behind occlusions in range data. Conf. on 3D digital imaging and modeling (3DIM), Montreal, Canada, pp. 232-239, 2001.
- [Dic01a] Dick, A. R., Torr, P. H. S., Ruffle, S. J. and Cipolla, R. Combining single view recognition and multiple view stereo for architectural scenes. In Int. Conf. Computer Vision, pp. 268-280, 2001.
- [Fis81a] Fischler, M.A. and Bolles, R.C. Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography. Communications of the ACM, 1981
- [Har03a] Hartley, R., Zisserman, A. Multiple View Geometry in Computer Vision, Cambridge University Press, ISBN: 0521540518, 2003.
- [San99a] Sanchiz, J.M., Fisher, R.B. A next-best-view algorithm for 3D scene recovery with 5 degrees of freedom, BMVC, pp. 163-172, 1999.
- [Wer02a] Werner, T. and Zisserman, A. New techniques for automated architectural reconstruction from photographs. European conference on computer vision, Copenhagen, Denmark, pp. 541-555, 2002.
- [Wer02b] Werner, T. and Zisserman, A. Model selection for automated reconstruction from multiple views. British Machine Vision Conference, Cardiff, UK, pp. 53-62, 2002.

# Viewpoint Selection Based on Fechner Type Information Quantities for 3D Objects

Satoshi.Oba<sup>1</sup>, Takeo.Ikai<sup>1</sup>, Shigeki.Aoki<sup>2</sup>, Takuya.Yamashita<sup>3</sup>,  
Masao.Izumi<sup>1</sup> and Kunio.Fukunaga<sup>1</sup>

<sup>1,2</sup>Osaka Prefecture University

1-1 Gakuen-cho, Nakaku, Sakai,  
Osaka 599-8531, Japan

<sup>3</sup>Panasonic Mobile & System Engineering Co.,Ltd.

1-2-33,Miyahara,Osaka,Yodokawa-ku,  
Osaka,532-0003 Japan

<sup>1</sup>{ ooba@com.cs, ikai@cs, izumi@cs fukunaga@cs } .osakafu-u.ac.jp

<sup>2</sup> shigeki\_aoki@m.ieice.org, <sup>3</sup> yamashita.takuya002@jp.panasonic.com

## ABSTRACT

This paper proposes several algorithms for selecting viewpoints, based on information quantities, which provide representative views expressing a whole feature of 3D object. By defining a novel information quantity of Fechner type based on Fechner's law in psychophysics, we introduce shape information quantities depend on an area of face and depend on a length and sharpness of edge line in a polyhedral object. We then define viewpoint information quantities of several types obtained by summing up shape information quantities of the visible surface form a viewpoint. Representative views are obtained from viewpoints at local maximum of the viewpoint information quantity of each type. The face type and the edge type of algorithms are derived that compute viewpoint information quantities obtained from all visible faces and all visible edge lines respectively. Experimental results and estimation on polyhedral objects and triangular mesh representations of curved objects are presented.

**Keywords** Viewpoint selection, Fechner type information quantity, Shape information quantity, Viewpoint information quantity, Representative view, 3D object.

## 1 INTRODUCTION

Viewing a 3D object, we can obtain different features of the 3D object from different viewpoints. It is therefore important to select a good view (or viewpoint) that grasps a whole feature of 3D object and the viewpoint selection has various applications such as computer graphics, object recognition, data visualization, etc. Many previous approaches on the viewpoint selection, for examples [SK92, TFTN05, VFSH01], are 2D image based approaches using information taken from 2D image such as a projected area of object surface. A 3D model based approach, on the other hand, is available which searches for good viewpoint using the 3D model such as a surface model, a wire-frame model [KK88], and so on. An advantage of 3D model based approach is that it allows us to use 3D information lost in projected 2D images. One of such lost 3D information is a curvature-like feature of surface, which is ob-

tained by the distribution of normal vectors of surface of a 3D model.

In pioneering researches [SK92, VFSH01] on general viewpoint selection, Shannon's entropy is utilized as an information measure to obtain good viewpoints in 2D image based approach. We however adopt a more natural approach for visual perception and introduce a novel information measure based on Fechner's law [P99] in psychophysics, which expresses the logarithmic characteristic of sensory response to stimulus.

In this paper, several algorithms for good viewpoint selection are proposed, which are applied from simple polyhedral objects seen in daily life to complicated polyhedrons containing polygonal expressions of 3D curved objects, by defining shape and viewpoint information quantities of Fechner type based on the 3D model based approach. The shape information quantity is defined based on a sensory amount obtained from an area of face or a length and sharpness of edge line of a polyhedron object, under the assumption such that a viewpoint (i.e. an eye) receives the light stimulus equivalent to the area of face or the length of edge line. Several viewpoint information quantities are then computed by proposed algorithms carried out various summations, which gather shape information quantities of visible surface of the object from a viewpoint. For viewpoint selection, these algorithms propose represen-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Copyright UNION Agency – Science Press Plzen, Czech Republic

tative views (or viewpoints) given by local maximum values of each viewpoint information quantity. We then suppose that there exists a good view for user in the set of representative views. In the derivation of algorithms, we suppose a virtual 3D model of object having the virtual surface constructed from sensory amount (i.e. shape information quantity) instead of the real physical surface of object. This virtual model may be considered as an internal 3D model (3D object image) in the brain.

The remainder of this paper is organized as follows. Section 2 refers to related work. The information measure of Fechner type is defined in Section 3. Two main types of algorithms are derived for viewpoint selection of polyhedral objects. We then construct a face type algorithm in Section 3 and construct edge type algorithms in Section 4. These algorithms are applied to polyhedral objects and triangular mesh expressions of curved objects that are regarded as complex polyhedrons. Section 5 is devoted to experimental results and estimation. Finally, in Section 6, conclusions are presented.

## 2 RELATED WORK

Shannon's entropy is applied in various field of information processing and computer vision. Takeuchi and Ohnishi [TO98] expressed the intensity information of 2D image by Shannon's entropy and proposed an active vision system finding complex region in a 2D image. Shannon's entropy is also introduced for viewpoint selection. Sato and Kato [SK92] defined the object image entropy to yield a good viewpoint with a balanced distribution of visible faces in a projected 2D image. Vazquez et al. [VFSH01] formulated independently the viewpoint entropy, which is the same as the object image entropy, except for adding to it the projected area of background in a 2D image.

Various applications of viewpoint entropy are carried out, such as image-based modeling [VFSH03], perception based illumination design [VS03], and volume visualization [TFTN05]. Weinshall and Werman [WW97] asserted two important measures such as the view likelihood, which is the probability for obtaining a characteristic view, and the view stability, which implies the stability of good viewpoint. Kamada and Kawai [KK88] provided, based on a 3D wire-frame model, the viewpoint obtaining cleared frame model without overlapping its edge lines using the normal vector of face from 3D model. Similarly using 3D models, Lee et al. [LVJ05] introduced the idea of mesh saliency applying mesh simplification and viewpoint selection.

Palmer et al. [PRC81] investigated the canonical view (or viewpoint) for an object using psychophysical measurements, which is assigned the highest goodness view and is first imagined in visual imagery of an object by people. Blanz et al. [BTB99] further investigated properties of canonical views for various objects including nonsense objects using computer graph-

ics psychophysics. One of purpose of viewpoint selection techniques is to obtain good approximation of the canonical viewpoint for an object. Although visible shape of 3D object varies infinitely according to the viewpoint transference, human has qualitatively limited and stable views about the object in spite of viewpoint transference. This cognitive fact is explained by the concept referred as the view potential in psychology [RF86].

## 3 FACE TYPE ALGORITHMS FOR VIEWPOINT SELECTION

There exist 3D model representations for a polyhedral object such as the surface model and the wire-frame model based on faces and edge lines respectively. The entropy method uses 2D projected image of the surface model. According to these two models, we construct a face type and an edge type of algorithms for viewpoint selection.

### Entropy Method and Assumptions

The entropy method for viewpoint selection of polyhedral objects such as the object image entropy [SK92] and the viewpoint entropy [VFSH01] is based on the assumption such that a good viewpoint depends on the largeness of number of visible faces and the uniformity of each visible area in given 2D image. Let  $T(z)$  be a set of visible faces from a viewpoint  $z$ . This entropy, denoted by  $H(z)$ , is obtained from the visible area  $A(t)$  of face  $t (\in T(z))$  as follows.

$$H(z) = - \sum_{t \in T(z)} \frac{A(t)}{\tilde{A}(z)} \log_2 \left( \frac{A(t)}{\tilde{A}(z)} \right) \quad (1)$$

where  $\tilde{A}(z) = \sum_{t \in T(z)} A(t)$ . In case of the viewpoint entropy,  $T(z)$  contains the projected area of background in given 2D image. Takahashi et al. [TFTN05] used for volume visualization a modified version of Eq.(1) by dividing it with its maximum value of  $\log_2 N$ , where  $N$  is the number of elements in  $T(z)$ . The value of  $A(t)$  is computed by counting the number of pixels belonging to the visible area.

In the problem of viewpoint selection, it is generally difficult to define an objective criterion on the good viewpoint, since it has qualitative properties depending on subject and sensibility, etc. We therefore stand on the assumption such that instead of qualitative property we introduce measurable quantity representing a difference of each viewpoint, in a set of maximal vales of which a good viewpoint exists. The face type algorithm is constructed based on the following assumption. (1) A necessary condition of good viewpoint is to capture the whole feature of a 3D object and therefore it is necessary to provide visible surface of the object as large as possible. (2) We look at the object with referring to its 3D image (3D object model possessed in the brain)

and search a good viewpoint into the brain such that it provides visible surface of this 3D image as large as possible. For realizing the above-mentioned assumption we introduce a novel information measure based on Fechner's law [P99], which is a more natural measure for visual perception than mathematical measure based on the information theory.

### Fechner Type Information Quantity

Fechner's law expresses logarithmic characteristics of sensory organs, which is a natural information measure for sensory amount or response to light stimulus from face and edge. Let  $Q$  be stimulus amount and  $R$  be sensory response, then the Fechner's law is expressed as,

$$R = c \log \frac{Q}{Q_0} \quad (2)$$

where  $c$  is a constant factor and  $Q_0$  is the lower limit of stimulus amount. Based on this relation, information quantity of the Fechner type, denoted by  $I$ , is defined as follows.

$$I = \log_2 \left( \frac{q}{\gamma} + 1 \right) \quad (3)$$

where  $q$  ( $\geq 0$ ) denotes physical or mathematical quantity as local stimulus of object surface, and  $\gamma$  ( $> 0$ ) is a design parameter tuning the effect of  $q$ . Information quantity  $I$  takes a nonnegative value by adding 1 to  $q/\gamma$ .

### Face Type Shape and Viewpoint Information Quantities

Now consider a point  $p$  on an object surface the neighborhood of which is regarded as a face or a plane. An information quantity of face type at the point  $p$  is defined by letting the area  $S$  of its neighborhood be the quantity  $q$  of Eq.(3). We stand on the assumption such that the viewpoint looking at the neighborhood of point  $p$  receives light stimulus equivalent to the area  $S$ . This information quantity of face type determines the information received by the viewpoint at infinity that is perpendicular to the face or the plane. Thus the shape information quantity of face type, denoted by  $I_f$ , is expressed as

$$I_f = \log_2 \left( \frac{S}{\gamma_f} + 1 \right) \quad (4)$$

Based on this shape information quantity, a viewpoint information quantity is introduced which a viewpoint receives from the visible surface of object. The viewpoint information quantity gathers shape information quantities of visible surface according to the viewpoint location and direction of surface. As shown in Fig.1, a viewpoint  $z$  is defined on the viewpoint hemisphere that a 3D object is set at its origin.

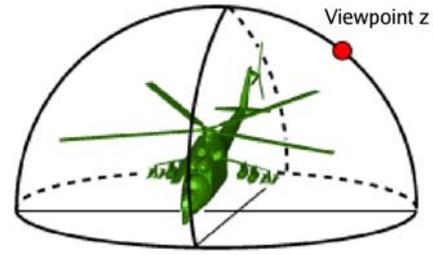


Figure 1: The viewpoint hemisphere

Let  $\zeta(t,z)$  be an angle between the normal vector of face  $t \in T(z)$ , where  $T(z)$  is a set of visible faces, and the view direction vector toward viewpoint  $z$ . Let  $S(t)$  be the area of face  $t$ . The step function  $g(x)$  is defined as  $g(x) = 1$  (if  $x > 0$ ),  $= 0$  (if  $x \leq 0$ ). If  $g(\cos \zeta(t,z)) = 1$ , then a face  $t$  is visible from the viewpoint  $z$ . An information quantity denoted by  $\Gamma(z,t)$ , that the viewpoint  $z$  receives from a face  $t$ , is expressed as,

$$\Gamma(z,t) = g(\cos \zeta(t,z)) \zeta(t,z) \log_2 \left( \frac{S(t)}{\gamma_f} + 1 \right) \quad (5)$$

when the viewpoint  $z$  is just above the face  $t$  (i.e.  $\zeta(t,z)=0$ ),  $\Gamma(z,t)$  has the maximum value and coincides with the shape information quantity  $I_f$  of Eq.(4). The viewpoint information quantity of face type, denoted by  $\Gamma(z)$ , is defined as the summation of shape information quantities, which the viewpoint  $z$  receives from all of its visible faces. Thus the information quantity  $\Gamma(z)$  is expressed as follows.

$$\Gamma(z) = \sum_{t \in T(z)} g(\cos \zeta(t,z)) \zeta(t,z) \log_2 \left( \frac{S(t)}{\gamma_f} + 1 \right) \quad (6)$$

### Internal 3D Models

Face type viewpoint information quantity for a face is represented as  $\cos \zeta \log_2 (S+1)$  (where  $\gamma_f=1$ ). Since the amount of light from face is supposed to be equal to  $S$ , an eye (a viewpoint) physically receives the light stimulus equal to  $S \cos \zeta$ , i.e. the visible area. In the case of  $\log_2(S+1) \cos \zeta$ , however, a shape information quantity  $\log_2(S+1)$  is the amount of sensory response and not the amount of light. In this model, a face has this sensory amount  $\log_2(S+1)$  instead of physical quantity of area  $S$ . This object model therefore has the virtual surface of sensory amount instead of actual surface of object.

An assumption for this model is then mentioned as follows. This virtual 3D surface model is supposed to be an internal model (3D image) of object that is possessed in the brain. If the retina captures a 2D image of some real object, a 3D object image corresponding

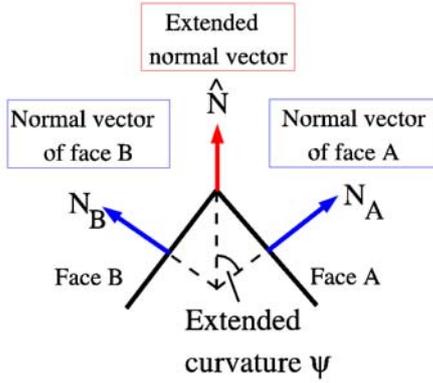


Figure 2: The extended normal vector and the extended curvature of edge line

to the retina image is called out in the brain. It is then supposed that the internal virtual viewpoint looks at this object image using the mental rotation in order to make matching with the retina image for the object recognition.

#### 4 EDGE TYPE ALGORITHMS FOR VIEWPOINT SELECTION

The edge type algorithm is constructed based on the following assumption. Taking notice of curvature-like quantity of an object surface we suppose that there exists more information at a surface having a lot of changes of curvature-like quantity, that is, an uneven and irregular surface has more information than even surface such as a face and a plane. Therefore the shape information for an edge line of polyhedral object is defined using its length and sharpness.

##### Edge type Shape Information Quantity

In order to define the edge type information quantity, we should represent the sharpness of edge line. The curvature of face is zero and that of line direction of edge line is also zero. There is however no ordinary curvature defined mathematically on the vertical direction of edge line. It is then necessary to introduce a curvature of wide sense.

So we define an extended curvature on the vertical direction of edge line. Fig.2 shows a section vertical to the edge line at any point of edge. An extended curvature is defined as 1/2 of the angle between the unit normal vectors  $N_A$  and  $N_B$  of faces A and B respectively which intersect at the edge line. This extended curvature is denoted by  $\psi$ . We next define an extended normal vector of edge line as the normalized vector of the sum of  $N_A$  and  $N_B$ , which is perpendicular to the edge line. A shape information quantity for an edge is then defined using the length of edge line and the inner product  $\cos \psi$  of the extended normal vector  $\hat{N}$  of edge

line and  $N_A$  or  $N_B$ . Thus the shape information quantity of edge type, denoted by  $I_e$ , is expressed as follows.

$$I_e = \log_2 \left( \frac{L}{\gamma_e \cos \psi} + 1 \right) \quad (7)$$

In Appendix, Eq.(7) is derived through the face type shape information quantity of Eq.(4).

##### Edge type Viewpoint Information Quantities

We now consider a classification of viewpoints for edges. When we look at an edge line, a neighborhood of edge line is also in sight. Two cases of viewpoint are therefore considered. One is (1) a viewpoint looking both faces that intersect at the edge line, and the other is (2) a viewpoint looking at least one of faces that intersect at the edge line. Viewpoint information quantities can be constructed corresponding to these cases. The cases (1) and (2) are called by the names as edge I and II types respectively in the following.

###### 4.2.1 Edge I Type

It is considered a situation that an edge line can be perceived and its sharpness (i.e. the extended curvature) is also recognized, only when both faces of the edge line are in sight. This situation may be considered as the case that a robot, which has no 3D image and knowledge about edge or object, perceives an edge line and its sharpness.

Let  $U(z)$  be a set of visible edge lines of object from a viewpoint  $z$  and  $\theta(u,z)$  be an angle between the view direction vector and the extended normal vector of edge line  $u$  ( $u \in U(z)$ ). Let  $\psi(u)$  be the extended curvature and  $L(u)$  be the length of edge line  $u$ . Moreover let  $\zeta(u,z)$  and  $\zeta'(u,z)$  be angles between the view direction vector and the normal vectors of both faces of edge line  $u$ , then both faces of edge line  $u$  are in sight if  $g(\cos \zeta(u,z))g(\cos \zeta'(u,z)) = 1$ . The edge I type viewpoint information quantity, denoted by  $\Omega_I(z)$ , is defined as the summation of shape information quantities, which the viewpoint  $z$  receives from all of its visible edge lines. Thus the information quantity  $\Omega_I(z)$  is expressed as follows.

$$\Omega_I(z) = \sum_{u \in U(z)} g(\cos \zeta(u,z))g(\cos \zeta'(u,z)) \cos \theta(u,z) \log_2 \left( \frac{L(u)}{\gamma_e \cos \psi(u)} + 1 \right) \quad (8)$$

###### 4.2.2 Edge II Type

Even though the case of viewpoint looking at least one of faces that meet at an edge line, the edge line is inferred at a margin of face and its sharpness is also speculated by the appearance of surrounding at the edge

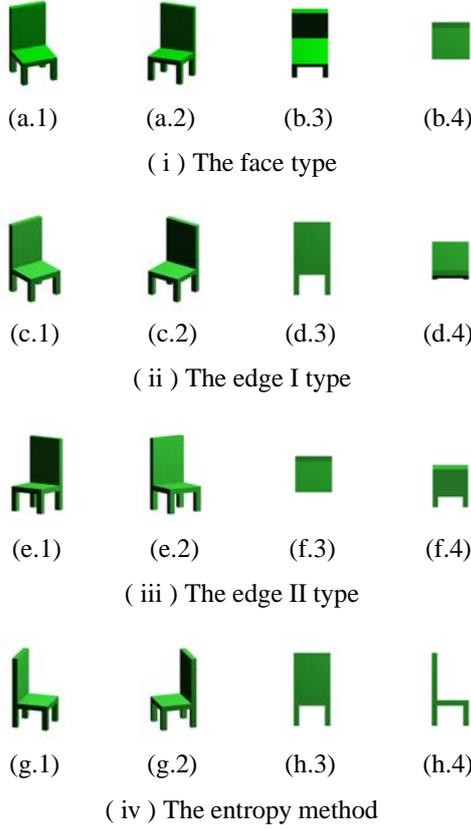


Figure 3: The 8 views of (a),(c),(e), and (g) are representative views and the 8 views of (b),(d),(f), and (h) are unrepresentative views of a chair by the face type, the edge I type, the edge II type, and the entropy type respectively.

line. This situation implies the fact that human (or a robot), which already has a 3D image and knowledge about edges or the object, can infer and perceives the edge line and its sharpness from such viewpoint. Under the assumption that the viewpoint receives a half of shape information quantity  $I_e$  of Eq.(7) when either face or edge is seen from the viewpoint, the edge II type viewpoint information quantity, denoted by  $\Omega_{II}(z)$ , is expressed as follows.

$$\Omega_{II}(z) = \sum_{u \in U(z)} \frac{1}{2} \{g(\cos \zeta(u, z)) \cos \zeta(u, z) + g(\cos \zeta'(u, z)) \cos \zeta'(u, z)\} \log_2 \left( \frac{L(u)}{\gamma_e \cos \psi(u)} + 1 \right) \quad (9)$$

#### 4.2.3 Convex-Concave Information Type

The edge I, II and the face type have following problem. The normal vector of surface is ordinary defined to have the direction toward outside from surface and the curvature-like quantity, such as  $\cos \psi$ , has

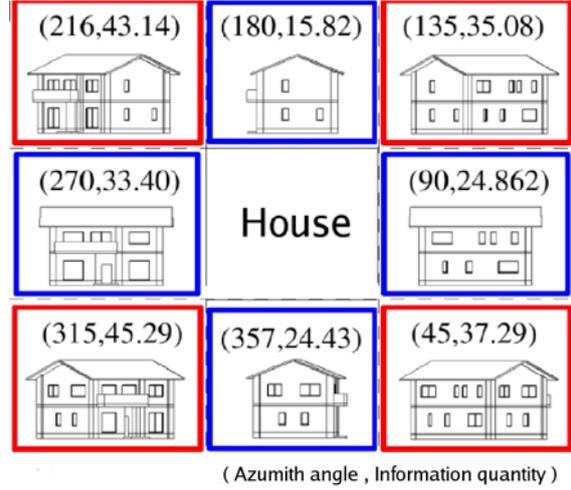


Figure 4: The view potential of a house

no sign. Faces and edge lines then cannot be determined whether they are convex or concave. Concave regions of surface are therefore regarded as convex regions, that is, the above-mentioned viewpoints are, as it were, equivalent to monocular stereopsis.

Edge lines are then classified using a method for deciding whether an edge line is convex or concave, and convex or concave information quantities are defined on convex or concave edge line respectively. The viewpoint information quantity  $\Omega_I(z)$  ( or  $\Omega_{II}(z)$  ) is therefore divided into convex and concave viewpoint information quantities denoted by  $\wedge^+(z)$  and  $\wedge^-(z)$  respectively. These information quantities are combined with a parameter  $\alpha$  ( $0 \leq \alpha \leq 1$ ) and yield the convex-concave information type, denoted by  $\wedge(z)$ , as follows.

$$\wedge(z) = \alpha \wedge^+(z) + (1 - \alpha) \wedge^-(z) \quad (10)$$

For above-mentioned viewpoint information quantities, that is,  $\Gamma(z)$ ,  $\Omega_I(z)$ ,  $\Omega_{II}(z)$ , and  $\wedge(z)$ , the viewpoint  $z$  at a local maximum value of each viewpoint information quantity is regarded as a representative viewpoint that is a candidate of viewpoint selection.

## 5 EXPERIMENTAL RESULTS AND ESTIMATION

This Section presents experimental results and estimation based on correlation characteristic for polyhedral objects and triangular mesh expressions of curved objects. The experiment uses the algorithm of detecting visible faces [MT97] for the triangular expression in order to make the set  $T(z)$ , and is carried out under 3600 viewpoints over the viewpoint hemisphere and  $\gamma_f = \gamma_e = 1$ .

### Polyhedral Objects

Various 3D objects having edge lines of right angles are found in daily life. Fig.3 shows experiment results for a

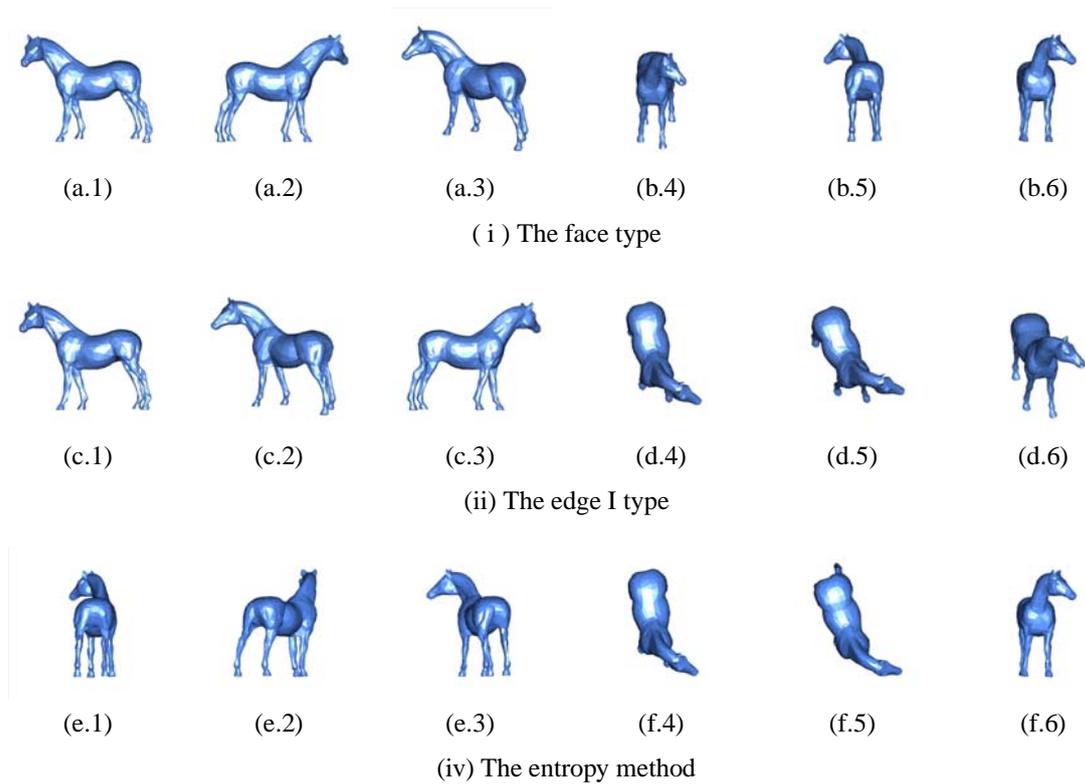


Figure 5: The 9 views of (a),(c),(e) are representative views and the 9 views of (b), (d),(f) are unrepresentative views of a horse by the face type, the edge I, and the entropy type respectively.

chair, which are viewpoints of the top two of local maximum values (representative views) and the last two of local minimum values (unrepresentative views) of each algorithm.

The view potential for a house is shown in [RF86] to be classified into 8 views around the house, which are coincident with views obtained by the edge II type viewpoint information quantity as showed in Fig.4. The views of four corners in Fig.4 surrounded by the red frame are representative views obtained from viewpoints of local maximum value and the other four views are obtained from viewpoints of local minimum value. In viewpoints of local minimum value there exist generally views (or viewpoints), which keep the shape features of the object (e.g. Fig.3 (b.3)) and do not keep that of the object (Fig.3 (b.4)). The viewpoint of minimum value looking at just above the roof of house does not keep that of the house. The keeping of the shape features is thus a necessary condition of view potential.

### Curved Objects

The polygon expression such as a triangular mesh expression for a curved object is a complex polyhedron and then the face type and the edge type algorithms can be applied to the polygonal expression. Fig.5 shows viewpoints of the top three of local maximum values (representative views) and the last three of local minimum values (unrepresentative views) of each algorithm

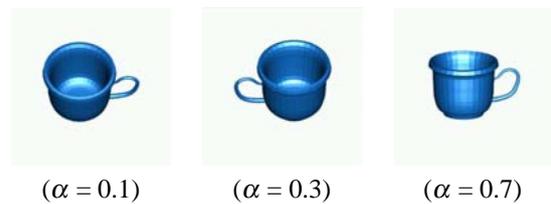


Figure 6: The representative views of a cup which are given by the convex-concave information type.

for a horse ( 1850 meshes ). In this experiment, the entropy method is computed by the triangular mesh 3D model of horse using  $A(t) = S(t)\cos \zeta(t,z)$  for  $A(t)$  of Eq.(1), where  $S(t)$  is the actual area corresponding to  $A(t)$ . An experimental result of the convex-concave information quantity  $\wedge(z)$  is shown in Fig.6. Functional viewpoints finding out a functional feature of object are obtained. As obvious from the figure, these viewpoints give some views by which we come to look gradually at the bottom of cup according to the value of  $\alpha$ .

### Correlation Coefficients and Estimation

We define a map which displays the distribution of viewpoint information quantity over  $(\theta, \phi)$  where  $\theta$  and  $\phi$  denote the azimuth angle and elevation angle of viewpoint respectively. An example for a cat is shown in Fig.7. Using this map, correlation coefficients among

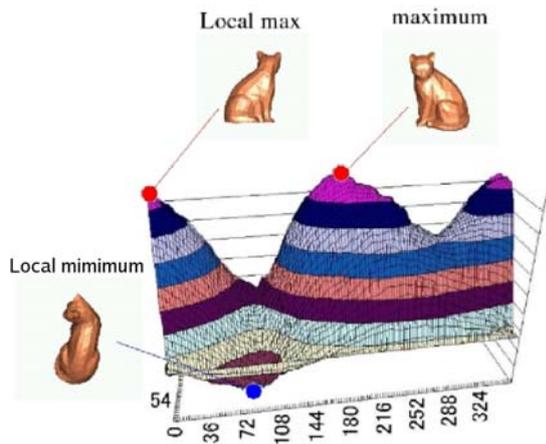


Figure 7: The map of viewpoint information quantity of the face type for a cat. Representative views at maximum and local maximum values and a unrepresentative view at local minimum value are shown.

maps of the algorithms for a chair and a horse are shown in table 1.

In the triangular mesh expression of curved object, large meshes and small meshes express small curvature parts and large curvature changes of a surface respectively. A area of triangle has a correlation with the sum of length of its edges and high approximation of curved object yields small extended curvature  $\psi$  and then  $\cos \psi$  nears to 1 since a change of normal vectors between adjacent meshes( patches ) is small. These conditions bring a high correlation between the face type and the edge type. This fact is shown in Fig.5 and the data of horse in Table 1. The edge II type is abbreviated in Fig.5 because it has very high correlation with the edge I type. The entropy method has however lower correlation with the others. The entropy grows larger at a part of uniform in size of meshes and this condition differs those of the face and the edge types.

On the other hand, for a polyhedral object the degree of correlation among the algorithms is slightly different for each other, witch is shown in Fig.3 and the data of chair in Table 1. Since all edge lines of chair have same right angles,  $\cos \psi$  of Eq.(7) is a constant value and then the correlation between the face and the edge types depends on that of areas of faces and lengths of edges. The entropy method however depends on the uniformity of visible areas of faces and then gives different views with lower correlation from other algorithms.

## 6 CONCLUSIONS

This paper presents several algorithms for viewpoint selection based on the assumption that there exists a good viewpoint for user's purpose in a set of representative viewpoints, obtained at local maximum of the viewpoint information quantity, which are supposed to

Table 1: Correlation coefficients among various types for a chair and a horse.

Chair	Edge I	Edge II	Entropy
Face	0.7284	0.7636	0.4718
EdgeI		0.8473	0.6478
EdgeII			0.6691

Horse	Edge I	Edge II	Entropy
Face	0.9331	0.9333	0.5133
EdgeI		0.9969	0.7753
EdgeII			0.7766

receive maximal amount of light stimulus from visible faces or visible edge lines determined by the viewpoint. Two main types of algorithms are derived based on faces or edge lines of a polyhedral object. The face type algorithm computes the viewpoint information quantity that is a sum of shape information quantities of all visible faces and the edge type algorithm similarly computes that of all visible edge lines from a viewpoint. Both types of algorithms give viewpoints with higher correlation for triangular mesh representations of curved objects and however provide different viewpoints with lower correlation for polyhedral objects. These characteristics of the algorithms shown for polyhedral objects are remained to investigate.

## 7 REFERENCES

- [BTB99] V. Blanz, M. J. Tarr, and H. H. Bulthoff, "What Object Attributes Determine Canonical Views", *Perception*, Vol. 28, No. 5, pp. 575-600, 1999.
- [KK88] T. Kamada and S. Kawai, "A Simple Method for Computing General Position in Displaying Three-Dimensional Objects", *Computer Vision, Graphics, and Image Processing*, Vol. 41, No. 1, pp. 43-56, 1988.
- [LVJ05] C. H. Lee, A. Varshney, D. W. Jacobs, "Mesh Saliency", *ACM Trans. on Graphics*, Vol. 24, No. 3, pp. 659-660, 2005.
- [MT97] T. Moller and B. Trumbore, "Fast, Minimum Storage Ray / Triangle Intersection", *Journal of graphics tools*, Vol. 2, No. 1, pp. 21-28, 1997.
- [P99] S. E. Palmer, "Vision Science", MIT Press, 1999.
- [PRC81] S. Palmer, E. Rosch, and P. Chase, "Canonical Perspective and the Perception of Objects", in *Attention and Performance Vol. 9*, pp. 135-151, Erlbaum, 1981.
- [RF86] I. Roth and J. P. Frisby, "Perception and Representation", Open University Press, 1986.
- [SK92] Y. Sato and T. Kato, "Information Amount for Polyhedral Images and Viewdirection Evalu-

ation” ,Trans. of IEICE, Vol J75-D-II, pp.1346-1352, 1992 (in Japanese).

[TFTN05] S. Takahashi, I. Fujishiro, Y. Takeshima, T.Nishita, “A Feature-Driven Approach to Locating Optimal Viewpoint for Volume Visualization”, Proceeding of IEEE Visualization 2005 pp.495-502, 2005.

[TO98] Y.Takeuchi and N.Ohnishi, “Active Vision System Based on Information Theory”, Systems and Computers in Japan, Vol.29, No.11, pp. 31-39, 1998.

[VFSH01] P.-P. V’azquez, M. Feixas, M. Sbert, and W. Heidrich, “Viewpoint Selection using Viewpoint Entropy” , VMV2001 pp. 21-23,2001.

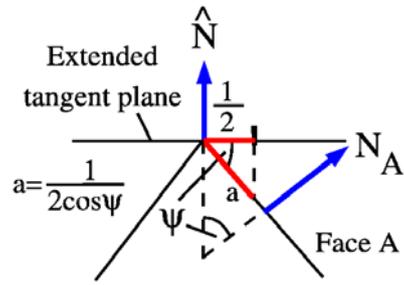
[VFSH03] P.-P. V’azquez, M. Feixas, M. Sbert, and W. Heidrich, “Automatic View Selection Using Entropy and its Application to Image-Based Modeling”, Computer Graphics Vol. 22, no.4, pp.689-700, 2003.

[VS03] P.-P. V’azquez and M. Sbert, “Perception-Based Illumination Information Measurement and Light Source Placement”, Lecture Notes in Computer Science, no. 2669, pp. 306-316. (Proc. of ICCSA’2003).

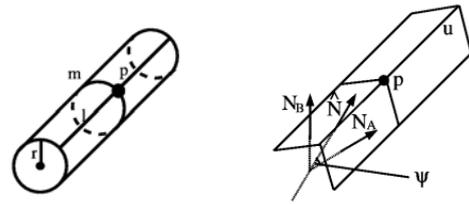
[WW97] D. Weinshall, M. Werman “On View Likelihood and Stability”, IEEE Trans. Pattern analysis and machine intelligence, Vol. 19, no.2, pp. 97-108, 1997.

## A DEERIVATION OF SHAPE INFORMATION QUANTITIES OF EDGE TYPE

As shown in Fig.A1, a plane, which contains edge line and is perpendicular to the extended normal vector  $\hat{N}$  at a point of edge line, is referred as an extended tangent plane of edge line. Let  $a$  be an area of some region in the face A, then  $a$  is viewed as the area  $a'$  ( $a' = a \cos \psi$ ) from the viewpoint at infinity which is perpendicular to the extended tangent plane. Suppose the visible area  $a'$  is  $1/2$  i.e. a half of unit area, then the actual area  $a$  is obtained as  $a = 1/2 \cos \psi$ . This quantity  $1/\cos \psi$  is regarded as the curvature-like quantity perpendicular to the edge line and the curvature-like quantity of edge line is 1 since its extended curvature is 0.



A.1 The extended tangent plane and the curvature-like quantity  $1/\cos \psi$



(a) the right cylinder

(b) the edge region of polyhedron

A.2 The correspondence between curvature  $1/r$  and extended curvature  $\psi$  of a right cylinder and an edge region of polyhedron respectively.

Thus the existence of two extended curvature on an edge region is shown as Fig.A2, based on the correspondence between an edge region and a right cylinder. A straight-line  $l$  and a circumference  $m$  of radius  $r$  are intersected at a point  $p$  over the surface of cylinder and they have curvature 0 and  $1/r$  as shown in Fig.A2(a). Similarly there exist the extended curvature  $\psi$  ( curvature-like quantity  $1/\cos \psi$  ) and straight-line of extended curvature 0 ( curvature-like quantity 1 ) at a point  $p$  over edge line shown in Fig.A2(b).

There exists a neighborhood of edge line whose area is  $L/\cos \psi$  that is the product of  $1/\cos \psi$  and length  $L$  of edge line. This neighborhood of area  $L/\cos \psi$  is in sight from the viewpoint above edge line. By substituting the area  $L/\cos \psi$  into Eq.(4), the shape information quantity of edge line is derived as follows.

$$\log_2 \left( \frac{L}{\gamma_e \cos \psi} + 1 \right) \quad (A.1)$$

# Measuring and segmentation in CT data using deformable models

Václav Krajiček

Faculty of Mathematics and Physics  
Charles University, Prague  
Malostranské nám. 25, 11800 Praha 1,  
Czech Republic  
vajicek@volny.cz

Josef Pelikán

Faculty of Mathematics and Physics  
Charles University, Prague  
Malostranské nám. 25, 11800 Praha 1,  
Czech Republic  
Josef.Pelikan@mff.cuni.cz

Martin Horák

Department of Radiology,  
Bulovka Hospital, Prague  
Budínova 2, 180 81 Praha 8 - Libeň,  
Czech Republic  
martin.horak@volny.cz

## Abstract

Accurate measuring of physical properties of human body has great importance for determining the best treatment. Our work aims at measuring volumes of organs such as kidney or liver in image data obtained from computed tomography (CT). We take advantage of long-time research in the area of deformable models. We have developed parametric model using closed B-Spline curves and have formulated energetic equation for their iterative evolution. Interior and exterior intensity distributions are taken into account, together with upcoming shape and position of regions in neighbouring slices of multi-slice CT data. This approach does not require gradient information, which is unreliable in medical images.

**Keywords:** Deformable Models, Active Snakes, CT, Medical Segmentation, Volume Measurement.

## 1 INTRODUCTION

Computed tomography is a common tool for medical diagnostics. It simply produces images of a human body interiors. They serve in the same way as X-Ray images have done for a century. They are as badly readable as X-Rays for untrained observer. On the contrary, CT images offer much more information. They are based on calibrated 3D data (position and density), organized into slices in axial plane of human body. Typical example of CT image can be seen in Figure 1. Thanks to these features we can use them not only for rough idea of human interior, but also for precise measurements of volume.

Physical measures of human body can say a lot about man's health. Remember, that temperature of healthy body is about 36,5 °C. Higher temperature means that body is fighting with an illness. The same goes for volumes of organs. Noticeable changes up or down of kidney-volumes indicate their proper function or dysfunction. These changes can be observed after a long period between CT scannings and also depend on correct and objective evaluation of CT images.

We are trying to develop methods for fast, accurate and automatic kidney segmentation which is precondition for measuring the volume. Nature of CT data and variety of human body make this task rather difficult.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.  
Copyright UNION Agency – Science Press, Plzen, Czech Republic

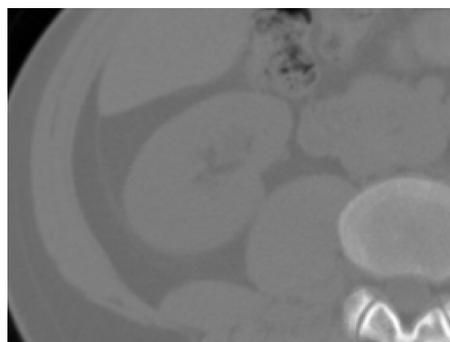


Figure 1: Slice of CT data from abdominal area.

## 2 RELATED WORK

Deformable models have been used and studied in computer vision for almost two decades. Their best known application is called 'active snakes' introduced in late 80's by Kass et al [KWT88]. They were based on evolving curves towards lowest total energy value defined mostly by image gradient and some regularization properties.

Unfortunately, medical images are very noisy and have unreliable gradient information. Chan and Vese [CV01] have introduced image energy term which depends on region statistics only.

$$E_{region1}(s) = \alpha \int_{int(s)} (f(x,y) - \mu_{ext}(s))^2 dx dy + \beta \int_{ext(s)} (f(x,y) - \mu_{ext}(s))^2 dx dy \quad (1)$$

Where  $s$  is closed curve,  $f$  is image and  $\mu$  is mean intensity of exterior an interior area we want to separate. In this case  $\mu$  can be known a priori or computed from initial position and updated during evolution.

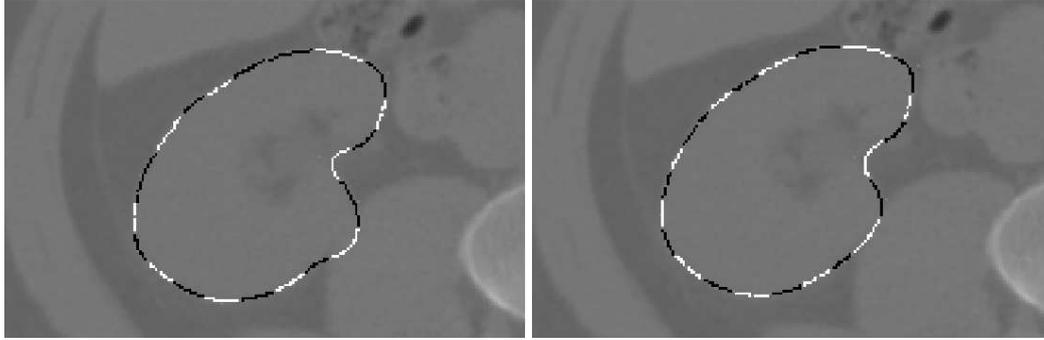


Figure 2: *left*: Results of converged B-Spline snakes without similarity force included. *right*: The same example of converged snakes with similarity forces included using parameter  $\gamma = 0.3$ .

We are using another improvement - classical curve representation is replaced by set of so called 'snaxels' with closed B-Spline curve. B-Spline curve as perfect tool for contour detection was introduced in [BHU00]. This approach reduces number of optimized parameters to coordinates of control points only. Using B-Spline implicitly brings benefit of intrinsic regularization property. B-Spline snakes with a very efficient computational scheme are presented in work of Jacob et al [JBU04].

### 3 ENERGY MODEL

We utilize a flexibility of snakes framework and we tried to find new terms which can be incorporated into it. We based our energy model based on regions. Contours in our model are represented by B-Spline curves.

In our first attempt we extend formula by an additional term which describes relation to the adjacent slices in CT dataset. This relation is based on a shape similarity measure. We are trying to find the best measure which is able to reduce unwanted behavior of the evolving curve.

$$E(s) = E_{region1}(s) + \gamma \cdot similarity(s, s_{neighbour}) \quad (2)$$

We tried to find how to express local high frequency changes of a shape in the similarity measure. This criterion can indicate overflow of the contour to a adjacent area through 'bridges' of similar intensity. These bridges should be narrow enough to be rendered as high frequency. High frequency can rise the similarity term in our formulation (2). Minimization scheme can isolate and eliminate them because of their significant differences. We construct shape describing vector as Fourier power spectrum of boundary orientation angle changes. Similarity of two shapes is equal to  $l_2$ -distance of their shape vectors. It is used as *similarity* term in (2). In [LAL03] it was shown that this similarity measure is sensitive to significant differences of corresponding parts of curves.

Simple Chan and Vese region energy scheme (1) incorporates square distance from mean, which is very

rough approximation of inverted probability. Instead of this we are directly using probability of element classification to exterior or interior region in fashion of [JBU04]. Furthermore we simplify weight coefficients  $\alpha$  and  $\beta$  to  $\alpha$  and  $(1 - \alpha)$  expressing balance between exterior and interior. It has only little importance in the new scheme.

$$E_{region2}(s) = -\alpha \int_{int(s)} \log(P_{int}(f)) - (1 - \alpha) \int_{ext(s)} \log(P_{ext}(f)) \quad (3)$$

Probability in (3) can be approximated by normal distribution from analysis of initial shape position. We can also completely rely on user's initial input and use normalized and optionally smoothed histogram of areas (interior, exterior) as our probability distributions. There are other interesting region-based energy models introduced in [JTW99].

$$E_{region3}(s) = -\frac{1}{2}(\rho_{int}^2 - \rho_{ext}^2)^2 \quad (4)$$

$$E_{region4}(s) = -\frac{1}{2}(\mu_{int}^2 - \mu_{ext}^2)^2 \quad (5)$$

Minimization itself can be achieved by the gradient descent method. Partial derivatives of functional  $E$  by parameters  $\mathbf{x}$  of the contour are the only terms to compute.

$$s_{n+1} = s_n + \delta \cdot \nabla_{\mathbf{x}} E(s(\mathbf{x})) \quad (6)$$

These parameters are in our case identical to control points of B-Spline curve. The most important feature of the region-based snake segmentation is possibility of reducing computational complexity from computing an area integral to computing a curve integral using Green's theorem.

We did not follow an example of [CV01] adding more than two types of energy functionals. It encounters

problem of energy equivalence which can not be always completely solved by parameters tuning. We can not simply (in meaning of linear relation) answer a question of how much region energy is equivalent to a unit of gradient based edge energy, length or area energy.

## 4 B-SPLINE CURVES

B-Spline curves are well known splines using polygonal base functions with limited support, which gives them great local control property. There are few problems we should handle with.

First is that we must use B-Splines in all energy functionals in place of a contour  $s$ . If we work with points on the contour and their displacement during the minimization step we must use control points only. It is a big advantage because there are fewer of them than contour points even if sampling is very rough.

First one - we must use B-Splines in all energy functionals in place of a contour  $s$ . If we are working with points on the contour and their displacement during the minimization step we must use control points only. This is a big advantage because there are fewer of them than contour points, even if sampling is very rough.

Order of B-Spline is another question. Cubic B-Spline is good enough for our purpose.

Number of control points or number of B-Spline sections depends on length of the whole closed contour. We define minimal and maximal length of segment and during iterations we check the length. If our limits are exceeded we split segment or merge adjacent segments as necessary.

The last problem relates to intersections. Sometimes evolving curve happens to be self-intersecting. For example if we are segmenting shape with holes, curve always intertwines around the hole and cycles forever. One solution consists in detecting intersections and breaking off parts of the curve. Free ends of the bigger part of the curve will be stucked together and the rest can be thrown away. We could also use the rest of the curve as an inner structure segmentation. This can be seen when segmenting higher or lower parts of kidney, which has pelvis in the middle (pelvis has very different tissue density). We might ignore it first time and later we can segment it by simple thresholding. Intersection detection can use convex wrapper property of B-Spline segments for computation speedup.

In [JBU04] they use integration of angle changes which gives various multiples of  $2\pi$  depending on number and direction of loops. This method is detecting loops which implicate self-intersections, but is ineffective in cases described above.

## 5 SEGMENTATION PROCEDURE

We are trying to segment 3D volume using separate segmentation of each slice. Shape similarity measure is used as inter-slice relation which binds adjacent slices

together. Reasons for our preference to this scheme (compared to pure 3D deformable model technique like Active Surfaces) are inhomogeneous data and computational efficiency which is far better in the case of 2D techniques. We are working with datasets of voxels which are larger in one dimension than in other two. In case of pure 3D technique, evolution in two different directions could be incomparable. This approach offers almost interactive response and better user control (better than editing 3D surface control points).

Our segmentation procedure starts specifying top and bottom end points of segmented organ by a user. Optionally user can set other points in slices between top and bottom ends, which can help to determine starting positions. Starting positions on other slices are placed automatically between these which were set manually by the user. Initial shape is defined as small circle around these points. Probability distribution is estimated from area inside these circles. It is important to put them in the right place inside segmented organ. The user can optionally supply probability distribution from manually segmented shape in one slice.

Iteration starts after shape initialization and placement on each slice. In each iteration of the procedure we perform one step of snake energy minimization. Thanks to this scheme the whole object is changing its shape during segmentation process and thus can be observed and possibly corrected by the user. After one of slices converges it became steady point for its neighbors and is available for use in similarity term in equation (2). Neighbors of already converged slices are restricted by similarity in their further evolution. This scheme works well for rounded organs whose axial slices do not change topology and shape variation too rapidly.

## 6 RESULTS

Convergence of the basic model (2) is strongly dependent on parameters  $\alpha, \beta, \gamma$ . The model (3), if supplied with proper probability distribution, gives very promising results. Our improvement with similarity measure can help to hold a curve in correct shape as can be seen in Figure 2.

We are working with native images of a kidney checkup and also with images of the same checkup with an contrast agent in several phases of saturation. In each case different strategy in parameter setting and probability distribution should be used to obtain optimal result. Native images work well with approximation by normal distribution. But in a case of thin patient lacking of contrast inter-organ fat, it is almost impossible to find a distribution which prevents overflowing to neighbouring organs without losing too much tissue on the surface of segmented organ. Using this scheme on slightly saturated kidney we are able to segment the whole organ and determine its volume well. Figure 4 shows final results

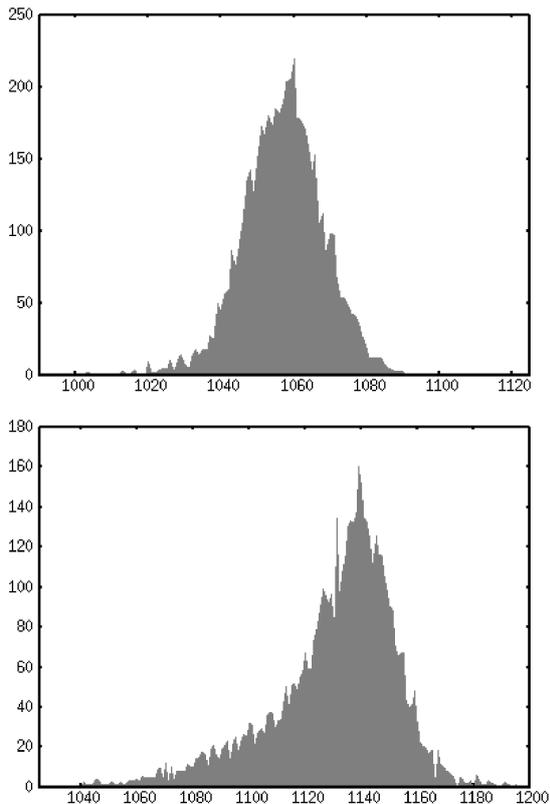


Figure 3: Histograms of segmented shapes. First graph from native CT corresponds to normal distribution. Second graph from an image with medium saturation of the contrast agent demonstrates slightly skewed distribution.

of our efforts. The approximation by normal distribution fails on a highly satiated kidney because it does not correspond to reality as shown in Figure 6. On the other side fine-tuned probability distribution works well on majority of patients.

A speed of the segmentation procedure is satisfying. Processing of a five-millimeter thick slices dataset takes about 80-90 seconds (intersection check is performed after each 40 iterations, convergence is achieved after about 400 iterations). On a two-millimeter slices dataset it takes 160 seconds, but the measurement is more precise. Time measurements were done on AthlonXP 2600+ computer.

## 7 CONCLUSION AND FUTURE WORK

We proved that our task can be effectively solved by segmentation using deformable models. Building efficient scheme for faster automatic segmentation will be our next goal. We have improved our framework by incorporating methods based on Green's theorem which speed up computations of region integral and its derivative. Detecting self-intersection is the slowest component of our algorithm, but also introduces simple topol-

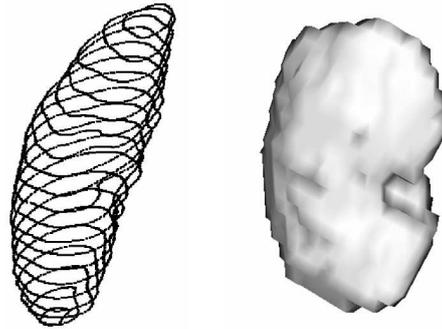


Figure 4: Schematic model(left) and 3D model(right) of segmented kidney.

ogy control (detection of holes). We are going to implement other enhancements like parallel computation of distant slices or multi-scale computations. Results of our measurements will be evaluated and assessed by medical specialist to determine its accuracy and possible usability in practice.

## ACKNOWLEDGEMENTS

This work is supported by GAUK grant No. 359/2006 (Charles University in Prague). MUDr. Martin Horák from Bulovka Hospital in Prague is providing us with all necessary data, medical knowledge and experience.

## REFERENCES

- [BHU00] Brigger, P., Hoeg, J., and Unser, M. (2000). B-Spline snakes: A flexible tool for parametric contour detection. *IEEE Transactions on Image Processing*, 9(9):1484–1496.
- [CV01] Chan, T. and Vese, L. (2001). Active contours without edges. *IEEE Trans. Image Processing*, 10(2):266–277.
- [JBU04] Jacob, M., Blu, T., and Unser, M. (2004). Efficient energies and algorithms for parametric snakes. *IEEE Transactions on Image Processing*, 13(9):1231–1244.
- [JTW99] Jr, A. Y., Tsai, A., and Willsky, A. (1999). A statistical approach to snakes for bimodal and trimodal imagery. In *ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2*, page 898, Washington, DC, USA. IEEE Computer Society.
- [KWT88] Kass, M., Witkin, A., and Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331.
- [LAL03] Lee, D., Antani, S., and Long, L. R. (2003). Similarity measurement using polygon curve representation and fourier descriptors for shape-based vertebral image retrieval.

# Comparative Navigation System for Collaborative Project

Oh, S.

National Institute of Information  
and Communications Technology  
3-5 Hikaridai,  
Seika-cho, Soraku-gun,  
619-0289, Kyoto, Japan  
sooyeon@nict.go.jp

Yeo, W.

Yonsei University  
134 Shinchon-dong,  
Seodaemun-gu,  
120-749, Seoul, Korea  
yeowh@yonsei.ac.kr

## ABSTRACT

We investigated the concepts, strategies, and functions of a 3D virtual design environment for collaborative, real-time architectural design using our 3D comparative navigation system and virtual reality technology. The development of the 'comparison' concept has enabled interactive design in real time in a 3D computer environment. Since participants must be able to easily understand the proposed design systems that help them gain this understanding are required. While comparison is an effective way to gain such an understanding, comparing one proposed design to another using existing systems is difficult because the user must operate their viewpoints separately. We therefore created a prototype system that displays different contents simultaneously while controlling the viewpoints automatically to facilitate content comparison. This comparative navigation system facilitates comparison of proposed designs by displaying related parts of the designs automatically. In this paper, we describe the concepts, strategies, and functions of a 3D virtual design environment for collaborative, real-time architectural design that is based on our 3D comparative navigation system and real-time simulation technology. We also evaluate the advantages and disadvantages of using this design environment for collaborative architectural design.

## Keywords

Collaborative design, Experience, Comparison, 3D Digital archive, Interactive, Real-time simulation.

## 1. INTRODUCTION

The use of 3-dimensional computer graphic (3DCG) models as an architectural design tool has been increasing in recent years. Awareness of design has been increasing, and awareness is essential for efficient architectural rendering and agreement between participants. Both specialists and non-specialists can more effectively understand a design by using 3DCG.

The inspection of designs by participants and the demand for presentations are high. Various viewpoints need to be considered in collaborative design, not only the viewpoints of the enterprise body or designer. There is also an increasing demand for designers to be able to respond immediately to demands in presentations.

Collaboration up till now has involved the creation of still pictures or animations from a viewpoint assumed in advance. But it is difficult to guess all the assumptions for all the needs that the participants. Moreover, during collaboration responses cannot be immediately made to spontaneous needs.

Here we focus on a 3D real-time simulation engine as a design tool that solves these problems immediately. It would provide an effective, interactive, and rapid design platform. It could be applied to all stages of the design process and be used to check the designs at any time, anywhere, and at any stage. Its support of visualization and interactivity would enable good communication between the client and designer, thereby reducing misunderstandings. The real-time interactive previews it would enable should become a major part of the design process.

The rapid increase in the power of personal computers along with the drop in their prices has led to the migration of visual simulation and computer animation applications from expensive workstations to inexpensive PCs. We can now obtain faster rendering and higher quality results from PCs. The development of 3D real-time visual simulation has enabled the rendering of high-quality images at high speed.

Architectural design involves two major components, i.e., photorealistic and scenario scripting, which

enable participants to feel a greater sense of realism. In the field of entertainment, movie makers and video-game programmers are investing a great deal of economic and human resources in developing a good interactive interface, e.g., a 3D real-time simulation engine. The entertainment field is expanding very quickly. This field has grown considerably over the past few years, and hardware and software are approaching perfection.

The aim of our research was to develop a good, interactive 3D development platform. It arose from an urgent need for 3D real-time simulation techniques that could be used to produce better architectural designs. We therefore focused on applying a real-time simulation engine to architectural design.

### **Comparison for Collaborative Project**

The participants in a design project must understand the proposal on which they are working. Systems that help them gain this understanding are therefore required. Comparison is as an effective method for assisting understanding. For example, the representation of a person can be compared with that of an object to help the user better understand a certain object. By considering the differences, the user can recognize and understand an object. That is, the user compares and contrasts to clarify the areas with similarities and differences, enabling the user to recognize each similarity and difference. Therefore, we put considerable emphasis on comparison. A person can clarify the correlation, the effect, and the causal relation of objects by multilaterally comparing the objects. In addition, a person can deepen his or her understanding of each feature of the contents being compared, resulting in a better overall understanding of the whole.

Comparing one type of content to another using existing systems is difficult because users have to locate contents and observe them from separate viewpoints. With this in mind, we developed a prototype system that simultaneously displays various types of content while automatically controlling the viewpoints, enabling the user to easily compare them. Such a comparative navigation system facilitates comparison of design proposals by displaying their related parts automatically. Our 3D comparative navigation system is semantically a 3D extension of such a system. In the real world, people can walk through only one space, while in virtual space, a user can walk through many spaces at the same time. This is a key concept of our system, and it is intuitive and effective.

### **Digital Archive for Comparison**

Digital archives initially included only text and photographs, while they now include photographs, videos with text explanations, and 3-dimensional computer graphic (3D CG) models. In the architectural field, for example, the use of 3D digital models for computer-aided design/computer-aided manufacturing (CAD/CAM) has improved the efficiency of design, construction, and management. The 3D model can be of huge objects that one can walk around, such as places with historical architecture or archaeological sites. The digital archives described in this paper assume a 3D model, hence the term ‘3D digital archives.’

Many case studies reflecting the expansion of 3D digital archives have been presented at international conferences <sup>[1-4]</sup>. The main focus of these studies has been the construction of the archives. There have been few reports of research into how such archives can be used or how they can be experienced.

The main feature of 3D digital archives is that users do not simply see a flat 2D image of the contents out of context—they can see the contents from all angles in a natural setting. In addition, users can experience the contents as if they were in the real world. Moreover, they can run simulations and scientific investigations that would not be possible in the real world.

## **2. NAVIGATION FUNCTION FOR COMPARISON**

To gain an understanding by comparison, we need to consider the original purpose and objective first. The content to be used changes on the basis of the comparative purpose and comparative object.

We have been examining various types of comparisons used in design projects over many years. The comparisons were made using various media, such as real-time simulation, animation, and still pictures. The functions required for comparison navigation were extracted by examining the comparison techniques used in the design projects. To generalize the comparison techniques and make them suitable for sharing, we arranged the main contents used for the comparisons on the basis of two viewpoints.

- Comparison of the differences in existence, form, and size to reveal the identity, similarities, heterogeneity, and features of 3D structures.
- Comparison of the changes in an object over time to reveal the similarities and differences at each stage, thus giving a picture of the change process.

The functions of comparison navigation were considered based on these two viewpoints. Users can

compare different contents to help them understand particular content. By considering the differences, users can recognize and understand the content. They can also clarify the relationships, effects, and causal relations between different contents by comparing the contents multilaterally. In addition, the process of comparison makes it easier to identify different features of the content, encouraging greater understanding. The functions were arranged in accordance with the comparison purpose, the target content, the viewpoint setup, the viewpoint movement, the screen separation, and the content expression. Consequently, ten functions were identified for general comparison navigation; screen division comparison, photograph/model comparison, transparency change comparison, superposition comparison, model change comparison, camera viewing angle change, shadow display, comparison object insertion, measurement, and guide map display.

We developed a comparative navigation system that uses these functions and real-time simulation to facilitate interactive comparative studies of 3D architectural designs.

### **3. COMPARATIVE NAVIGATION SYSTEM**

In this section, we describe the development of our comparative navigation system for a 3D architectural design and our prototype system.

#### **Development**

We used various tools, including an authoring tool, a modeling tool, and an image-editing tool, to develop this system. The main technology used was real-time simulation technology based on virtual reality.

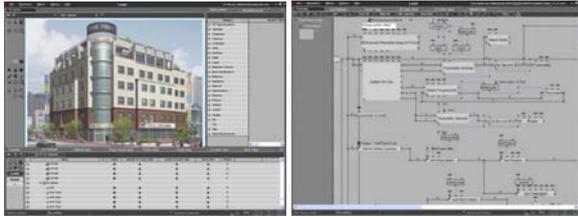
We used the Microsoft® DirectX Graphics-Application Programming Interface (API) based Virtools®, an authoring tool commonly used to develop computer games, to develop the internal scripts (Figure 1) comprising the real-time simulation graphical user interface (GUI). Producing architectural simulation is an impossible task for non-professional programmers, and architectural designers basically have no idea of how to produce a 3D scene. Virtools' building block system was specifically designed to meet the needs of cutting-edge interactive 3D development and is the only interactive 3D authoring tool accessible to non-programmers. Building Block is a subprogram packaged in a dynamic link library.

We assign building blocks in the scene as object behaviors through the visual authoring interface. We can then modify the scene by linking building blocks (using another subprogram). Specifically designed to meet the production needs of cutting-edge

architectural simulation, the building block system provides the groundwork and tools users need to unleash their creativity and harness the full potential of the 3D real-time simulation engine. Users can import industry-standard media files to the building block system as 3D models, textures, characters, sets, and sounds. They can attach behaviors to these entities to create interactions. They can control and tweak the behaviors to form a higher-level element that forms the foundation for interactivity or simulation. The building block system's intuitive GUI enables real-time 3D environments to be designed and instantly experienced in an interactive sophisticated manner. Behaviors can be collected from a multitude of sources (libraries, other projects, etc.) and be exchanged over the Web. This system's open architecture makes all the behaviors compatible, so they can be recombined with existing modules. The interface is constructed using Virtools® scripts, as shown in Figure 1.

Specifically, we used a note PC with a 4.3-GHz CPU, an ATI Radeon® 9800 GPU, and 2-GB RAM. To enable us to perform the rendering in real time with smooth movement, the rendering had to be done at no less than ten frames per second (FPS). This system was designed for comparative navigation in collaborative architectural design. The system had to meet three conditions in particular for it to support real-time simulation.

1. *High-speed rendering*: There is a trade-off between a high sense of reality and high-speed rendering as the system may not have sufficient performance for both. Priority was thus given to rendering at high speed to archive real-time simulation. The system also had to ensure the highest sense of reality.
2. *Lightweight 3D data*: One way to increase the rendering speed is by reducing the weight of the data. A balance needs to be found between sufficient data speed and a sense of reality. Moreover, the user should not feel impatient while using the system through the Internet. That is, the system should be able to read the 3D data in less than 90 seconds. This can be achieved by selecting suitable hardware and software.
3. *Easy operation interface*: The interface should be easy to operate. In situations where operation does not keep up with the rendering, the rendering speed should be reduced. Moreover, the interface should be immediately usable, even by a first-time user.

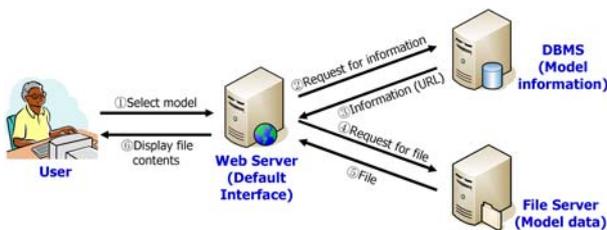


**Figure 1. Scripts of real-time simulation interface developed using Virtools®.**

## System Outline

This system enables a general user to construct a building in virtual space. The internal and external design models for the building are first recorded on the modeling server. If the name of the building that the user wants to access and use for the server for network distribution is already defined, interior and exterior space models suitable for it are retrieved from the modeling server. Moreover, if a design proposal to use it is already defined, the model is loaded from the modeling server and sent to the user. The user can design the proposal while manipulating the model in virtual space. The user can also record the data on the server using his or her ID. The system concept is illustrated in Figure 2.

1. The user accesses the web and selects a model from a menu.
2. The web server sends a request to the database management system (DBMS) for information about the requested model.
3. The DBMS sends the uniform resource locator (URL) of the file server containing the requested information.
4. The web server sends a request to the file server for the target file.
5. The file server sends the requested file. The web server displays the file contents to the user.



**Figure 2. System concept**

## Prototype

The prototype system supports a traditional walk-through simulation and is especially aimed at enabling users to experience the archive information through browsing in the following ways.

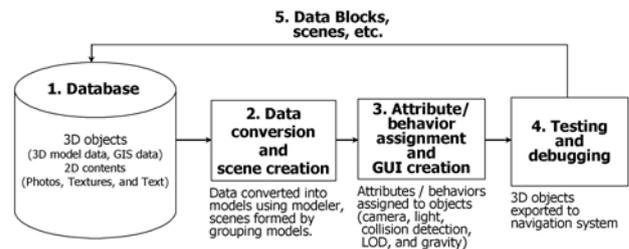
- As the user freely walks through the archives, the system provides on-demand comparative views of related content.

- If the user ‘collides’ with a model wall, a collision detection function generates a rebounding effect, similar to the impression received when colliding with a wall in an actual building.
- The user’s view is fixed at eye level by a ‘gravity’ function. The user does not ‘sink’ into the floor or ‘float’ above it but rather walks around as in the real world.
- The user can compare the various types of archive contents interactively. As soon as the user selects contents to be compared, the system displays them.
- The user can select from several interfaces—a mouse, a keyboard, a game controller, and a space/mouse traveler.

The process of making a comparative navigation system for a 3D architectural structure experience can be divided into five steps (Figure 3).

- Step1: Gather all data created by the modeling work; 3D models, GIS models, photos, etc.
- Step2: Convert the data into models using modeling tool. Then group the models together to form scenes within a circumference setting, a material setting, and alterable models.
- Step3: Assign attributes and behaviors to the objects and scenes in the setting, including the camera settings, light settings, collision detection settings, level of detail (LOD) settings, and gravity settings. Also create the GUI and system functions.
- Step4: Export the comparative navigation system for the 3D digital archive experience for testing and debugging.
- Step5: Save behavior blocks, scenes, etc. constructed during testing in the database.

The GUI operates as shown in Figure 4. The user can compare various types of 3D digital archive by switching between one-screen comparison, two-screen comparison, and four-screen comparison.



**Figure 3. Steps in creating comparative navigation system**

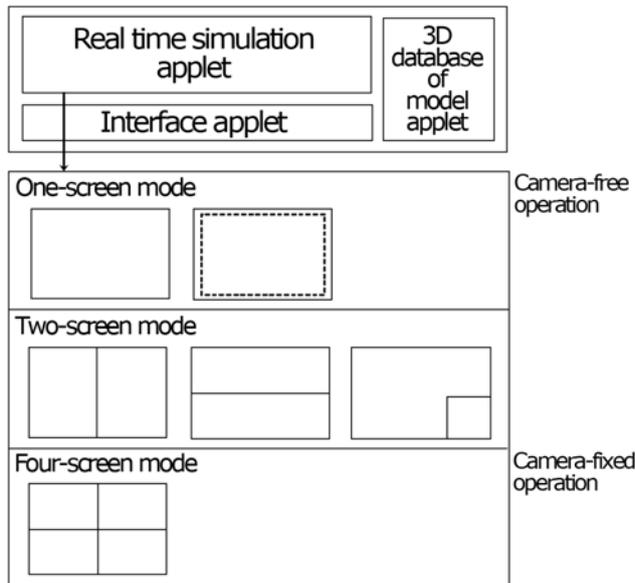


Figure 4. Operation of GUI

## Functions

This chapter will show the functions of the system. It will then present a method of improving a scene in a real project. In practice, we constructed a prototype system using the 3D digital archives of the historical architecture of a church on Gemiler Island in the Turkish Republic. This will be used to explain the functions of the system. The main functions of the system are 3D space move, plug-in, concurrent comparison navigation, and cross-section viewing function.

### 3.1.1 3D space move function

The basic functions for moving in 3D space are 'free walk' and 'free flight.' They enable a user to examine a proposed design in 3D space. The user can freely start at any point and move freely in 3D space.



Figure 5. 3D space move function

### 3.1.2 Plug-in function

Using a system like that illustrated in Figure 2, a user can choose the data insertion function from a menu on the screen. The system then loads a 3D model who the user chose from the data base for the design. The 3D model can be arranged freely. Moreover, the rearrangement is also possible after arrangement, rotation, and scale change.



Figure 6. Plug-in function

### 3.1.3 Concurrent comparison navigation function

Concurrent comparison navigation enables the user to comparatively examine a design proposal on one, two, or four screens by using easy key operations. In one-screen mode, the user can compare two proposed plans on the same screen. In two-screen mode, the user can compare two or more proposed designs by dividing the screen, as shown in Figure 4. In four-screen mode, the user can compare a proposed design with a photograph, etc.

#### 3.1.3.1 One-screen mode

- *Superposition comparison:* Position and size can be compared by layering two types of content and adjusting the transparency of one of the other, as shown in Figure 7.

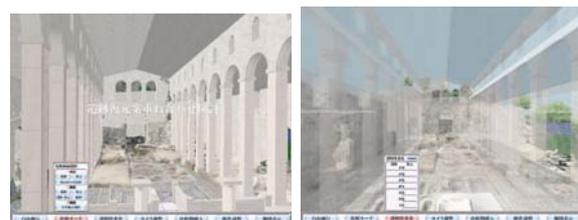
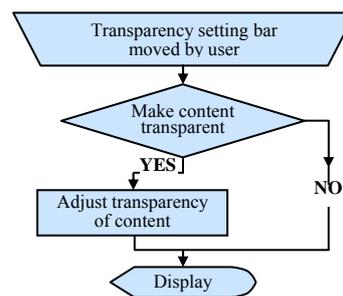
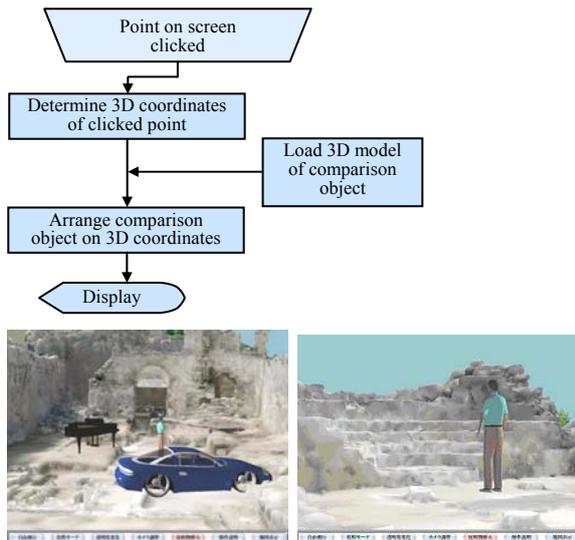


Figure 7. Superposition comparison

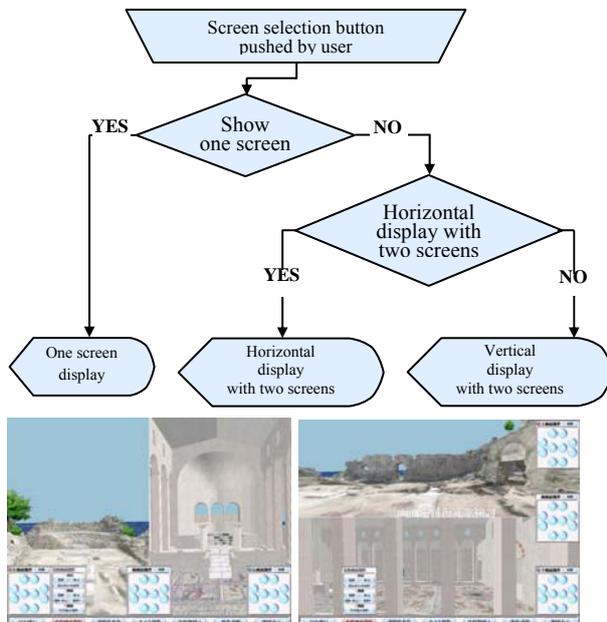
- *Object insertion comparison:* Size and scale can easily be grasped by inserting and displaying a 3D model of an object with a known size and scale, as shown in Figure 8.



**Figure 8. Object insertion comparison**

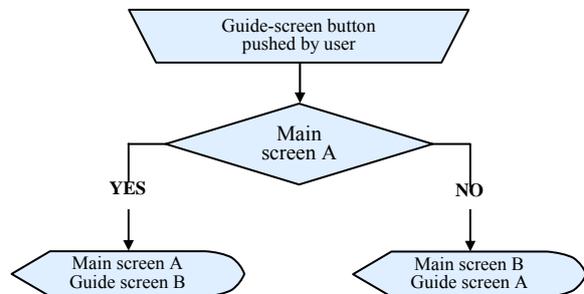
### 3.1.3.2 Two-screen mode

- *Two-screen comparison by vertical/horizontal screen division*: A user can compare contents while walking in a virtual space by displaying two types of content in two spaces on the same screen simultaneously, as shown in Figure 9. A camera and an aspect are defined for each space, enabling the user to better understand the space composition. A camera controller is displayed in each space, and the user operates it to adjust the view. A controller can also be displayed at the center of the screen for operating the two cameras simultaneously, enabling the user to traverse the same route in both spaces.



**Figure 9. Two-screen comparison**

- *Two-screen comparison with guide screen*: A third, smaller space can be added to show in more detail one of the two contents being displayed, as shown Figure 10. Using this ‘guide screen’ enables the user to focus on one of the contents and look at it in depth. The contents displayed in the corresponding main space can be changed to match those of the guide screen.



**Figure 10. Two-screen comparison with guide screen**

### 3.1.3.3 Four-screen mode

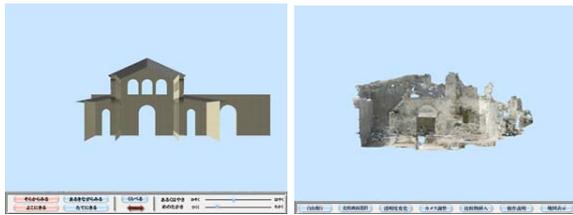
- *Photograph/model comparison*: The user can select a photograph from a photograph database (Figure 11, left), and the selected photograph is displayed in the lower left space (Figure 11, right). At the same time, models with the same viewpoint as the photograph can be displayed in the two upper spaces. Using this function enables the user to simultaneously compare the contents of various media, such as a 3D model of an excavation site, a 3D model of the restoration, and a photograph of a particular spot. It also makes it possible to position the camera at a particular viewpoint.



**Figure 11. Photograph/model comparison**

### 3.1.4 Cross-section view function

- The user can display a cross-sectional view of a structure by controlling the cutting plane, as shown in Figure 12. Using this function enables the user to understand the inner structure in detail.



**Figure 12. Cross-section viewing function**

#### 4. Prototype testing

We tested the prototype system during NICT's open house days in July 2004 and July 2005. The system was used by a total of 134 persons. The interfaces available were a keyboard, a mouse, a game controller, and a space traveler. Each user selected the one easiest for him or her to use. Next, like a person visiting a museum or cultural heritage site, the user was able to freely walk around in a 3D virtual space and experience a cultural heritage site. As the user freely walked through the 3D archives, the system provided comparative views of related content. For example, the user could experience the contents from a different time on the same screen.

Our prototype system worked smoothly, and the users could easily operate the navigation system. The children especially preferred using the game controller and were able to easily master the system, approaching it as if they were playing a game. They thus were able to experience the cultural heritage.

The users made several useful suggestions. For example, some suggested including more detailed explanations text would enable them to learn more, thereby enhancing the experiencing of a cultural heritage site using our navigation system.



**Figure 13. Photographs of users during open house**

#### 5. CONCLUSION

In conclusion, as the first step in developing a shared 3D environment that enables users to interact and understand architectural 3D models collaboratively, we have developed a prototype of an interactive navigation system that supports comparison. We have thus prepared the basic technology for experiencing architectural spaces by quickly examining designs through the Internet. Our comparative navigation system equipped with a 3D database makes it possible to use 3D data and comparison functions for various purposes in

architectural design. Furthermore, the interactive interface built into the real-time rendering system enables a knowledge-exchange architectural design system to be developed, thus providing alternatives to traditional architectural design systems. By using this system, users can experience 3D contents comparison by quickly examining contents through the Internet.

Testing at an open house showed that users ranging from children to senior citizens can easily experience digital archives using our prototype system. Our proposed system promotes the use of digital archives and content by enabling users to interact with the archives, thus raising their level of satisfaction. The development of this system has made it possible to use 3D archives for various purposes. This system is thus a crucial academic compilation containing architectural technologies and cultural aspects of historic architectural structures that have previously lacked clear academic definition.

The most serious problem in the development of a comparative navigation system is the need to process more than 700,000 polygons for creating a one-frame image. This is not possible with the hardware setup we used, which has a rendering speed of 0.3 FPS. We thus reduced the number of polygons to 100,000, which can be handled by the present hardware. This time, we divided the plane portions from the curved surface portions and used a polygon reduction algorithm to carry out the maximum maintenance of the present form. We also switched to the DirectX® rendering engine. Furthermore, we were able to achieve a rendering speed of 10 FPS by applying LOD and a clipping algorithm.

We plan to develop an improved version of our system as a trial production system. There are several improvements that can be made directly to the prototype system. First, the data format should be changed as VRML is not suitable for huge models of archaeological sites. Formats such as Web3D might be better. Converting the VRML format data into XVL format data would reduce the model size by about 30%. We also plan to enhance the user interface to enable it to handle other types of media. Furthermore, we are considering the development of a general-purpose function to support access to various types of 3D digital archives. In connection with this, additional 3D digital archives need to be created. Finally, we intend to develop tools that support an expression technique with sufficient reality and that limit the increase in the amount of data and to develop a system that enables smooth cooperation with the database.

## 6. REFERENCES

- [1] Beraldin, J.A., Picard, M., El-Hakim, S.F., Godin, G., Valzano, V., Bandiera, A. and Latouche, C. Virtualizing a Byzantine Crypt by Combining High-resolution Textures with Laser Scanner 3D Data. *8th International Conference on Virtual Systems and MultiMedia*, pp. 3–14, 2002.
- [2] Kalnins, R.D., Markosian, L., Meier, B.J., Kowalski, M.A., Lee, J.C., Davidson, P.L., Webb, M., Hughes, J.F. and Finkelstein, A. WYSIWYG NPR; Drawing Strokes Directly on 3D Models. *ACM SIGGRAPH*, pp. 755–762, 2002.
- [3] Pollefeys, M., Van Gool, L., Vergauwen, M., Cornelis, K., Verbiest, F. and Tops, J. 3D Recording for Archaeological Fieldwork. *IEEE Computer Graphics and Application*, pp. 20–27, 2003.
- [4] Schnabel, M.A., Kuan, S. and Li, W. 3D Scanning, Digital Modeling, Rapid Prototyping and Physical Depiction. *9th International Conference on Computer Aided Architectural Design Research in Asia*, pp. 227–237, 2004.
- [5] Tsuji, S. ed. The Survey of Early Byzantine Sites in Oludeniz Area ( Lycia, Turkey). *Osaka University*, Vol.35, 1995.
- [6] Chen, E.S. QuickTime VR – An Image-based Approach to Virtual Environment Navigation. *ACM SIGGRAPH Computer Graphics*, pp. 29–38, 1995.
- [7] Pfister, H., Zwicker, M., van Baar, J. and Gross, M. Surfels: Surface Elements as Rendering Primitives. *ACM SIGGRAPH*, pp. 335–342, 2000.
- [8] Tsukamoto, M. Image Based Pseudo-3D Visualization of Real Space on WWW. *Digital Cities: Technologies, Experiences, and Future Perspectives, Lecture Notes in Computer Science, Springer-Verlag*, Vol. 1765, pp. 208–302, 2000.
- [9] Sagawa, R., Nishino, K., Wheeler, M.D. and Ikeuchi, K. Parallel Processing of Range Data Merging. *IEEE/RSJ International Conference on Intelligent Robots and System*, Vol. 1, pp. 577–583, 2001.
- [10] Knight, M.W. and Brown, A.G.P. Navrgate X, Naturalistic Navigation Metaphor for Large Scale Virtual Environments. *8th International Conference on Computer Aided Architectural Design Research in Asia*, pp. 625–629, 2003.
- [11] Yeo, W., Lou, C.W., Kaga, A., Byun, C., Ikegami, T. and Sasada, T. An Interactive Digital Archive for Japanese Historical Architecture. *8th International Conference on Computer Aided Architectural Design Research in Asia*, pp. 513–522, 2003.
- [12] Tanaka, H. Deformable Maps and Spaces: Visualizing Users' Points of Interest and User Communication. *Second International Conference on Creating, Connecting, and Collaborating through Computing*, pp. 86–93, 2004.
- [13] Yeo, W., Yuda, Y., Oh, S., Kaga, A., Sasada, T. and Sato, R. The Development of an Environmental Design Helping System Using Real-time Simulationguide. *9th International Conference on Computer Aided Architectural Design Research in Asia*, pp. 947–955, 2004.
- [14] Barsoum, E. and Kuester, F. Web VR: An Interactive Web Browser for Virtual Environments. *International Conference on Electronic Imaging Science and Technology*. Vol. 5664, pp. 540–547, 2005.
- [15] Oh, S., Kadobayashi, R., Sasada, T. and Tanaka, K. Collaborative Project to Conserve and Personalize the 3D Digital Archives of Historical Architecture. *10th International Conference on Computer Aided Architectural Design Research in Asia*, pp. 179–187, 2005.
- [16] Palmon, O., Oxman, R., Shahar, M. and Weiss P. L. Virtual environments in design and evaluation. *International Conference on Computer Aided Architectural Design Future2005*, pp. 145–154, 2005.
- [17] Oh, S., Tanaka, K. and Sasada, T., 3D Digital Archive Experience for Historical Architectures, *International Conference on Education and Research in Computer Aided Architectural Design in Europe*, pp. 573–580, 2005.
- [18] Cortona VRML Client.  
<http://www.parallelgraphics.com/products/cortona/>
- [19] X3D. <http://www.web3d.org>
- [20] OpenGL. <http://www.opengl.org/>

# Using Graphics Hardware for Multiple Datasets Visualization

Gaurav Khanduja  
Louisiana State University  
Department of Computer Science  
USA (70803), Baton Rouge, Louisiana  
khanduja@bit.csc.lsu.edu

Dr. Bijaya B. Karki  
Louisiana State University  
Department of Computer Science  
USA (70803), Baton Rouge, Louisiana  
karki@csc.lsu.edu

## ABSTRACT

We have applied three graphics hardware-based approaches to support concurrent visualization of multiple sets of volumetric scalar data. They include volume rendering, clipping and isosurface extraction methods, which exploit 3D textures and advanced per pixel operations. These methods are expected to give better interactive frame rates for multiple datasets visualization (MDV) compared to the software-based methods. The rendering time in each case increases nonlinearly with the increasing the number ( $N$ ) of the datasets being visualized. We can identify three regimes, which can be characterized by different *time-N* slope value. The first regime with small slope value continues up to about 5 datasets, then the second regime with medium slope value continues up to about 25 datasets, and finally the third regime with much larger slope value continues up to 35 datasets. With volume shading enabled, the rendering time increases on average whereas the transition and maximum  $N$  values decrease. We propose the dynamic-resolution approach for increasing the maximum  $N$  and frame rates for above MDV techniques.

## Keywords

Multiple datasets visualization, 3D textures, GPU programming, Volume rendering, Clipping, Isosurface extraction, volume spreadsheets

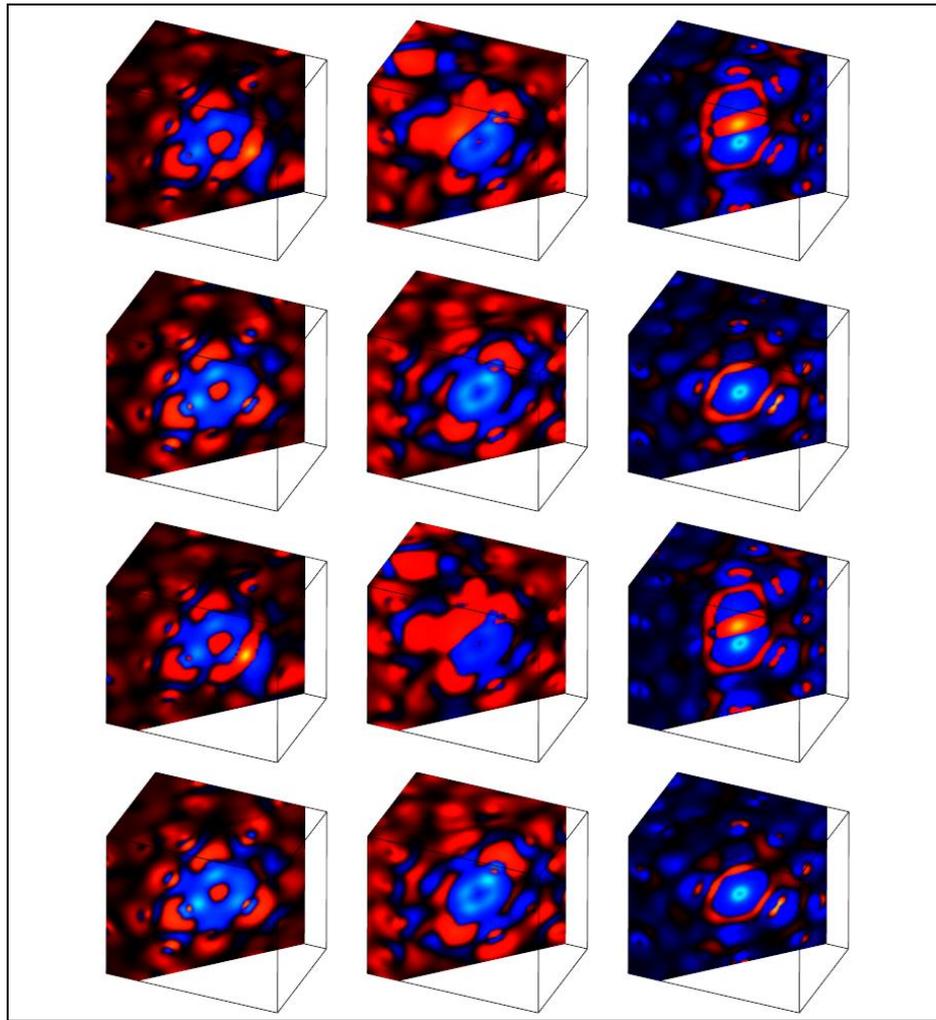
## 1. INTRODUCTION

Visualization of three-dimensional scalar data has played a crucial role in understanding the behavior of an associated system. However, often for complete insight in the physical system there is a need to analyze different datasets simultaneously of the given system, representing different time step, pressure, temperature. Multiple Datasets Visualization (MDV) [Sch04, Kha05, Kha06a, Kha06b] plays a crucial role in understanding such physical systems. MDV means, rendering two or more dataset in the same visualization either side by side or in fused form. In this paper, we focus on the MDV techniques in light of recent advances in GPU technology and texture based rendering using volume spreadsheets (side-by-side comparison of datasets). Khanduja and Karki

[Kha06a, Kha06b] showed the application of MDV for three dimensional scalar volume data. Similar concept for 2D images has been studied widely [Lev94, Chi97, Jan00] for over the decade. Figure 1 shows the visualization [Kar06] for magnesium silicate (post perovskite) data after clipping.

Primary goal is to achieve interactive visualization for volume scalar data. We present three texture-based approaches exploiting the features of the modern graphics hardware for MDV. The first one is volume rendering involving texture mapping slice by slice with appropriate alpha blending enabled [Cab94]. The second approach is clipping, to uncover important details of a dataset. The planar and box clipping [Kha06b] represents the simple form of clipping based on surface rendering. We also extend the idea of voxelized clipping [Wei03] to MDV. This technique requires an extra 3D texture representing the clip geometry. Third approach is the isosurface extraction using 3D textures. The method proposed here does not require the polygonal representation of the isosurface geometry and thus is applicable to MDV for improving interactive frame rates.. To further improve the effectiveness of above mentioned techniques, volume shading based on the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Copyright UNION Agency – Science Press, Plzen, Czech Republic.



**Figure 1: Visualization of electron density difference induced by the Mg (left), Si (centre) and O (right) vacancies in the 60-site  $\text{MgSiO}_3$  system. First and second rows show final configuration (after atomic structural optimization) for migrating ion and fixed vacancy respectively. Third and fourth rows show the corresponding initial configuration.**

phong model [Pho75] is applied in conjunction with the above texture based techniques.

The outline of the paper is as follows. In Section 2, related work is discussed. In Section 3, different texture-based MDV techniques including volume rendering, box clipping, voxelized clipping and isosurface extraction are presented. In Section 4, volume shading is discussed. In Section 5, we discuss dynamic resolution approach. Finally, the Section 6 contains important conclusions and future directions.

## 2. RELATED WORK

Visualization of three-dimensional scalar data has been area of research for over last two decades. Several visualization methods are available for volumetric scalar datasets [Mei00]. Multiple datasets have been visualized in many occasions [Cru96; Sch04, Kha05, Kha06a, Kha06b]. Schluzer [Sch04] uses concept of multiple variable of dataset; all time

steps are presented as animation and multiple datasets side by side for MDV. Khanduja and Karki [Kha06a, Kha06b] proposed isosurface extraction using Marching Cube [Lor87] algorithm and texture based approach for MDV. MDV graph (rendering time vs. number of datasets) in [Kha06a, Kha06b], shows non-linear behavior. Both the polygon generation time and the polygon rendering time [Kha06a] show the rapid increase once the swapping between main memory and virtual memory starts occurring due to increase in the number of datasets. Similarly, in [Kha06b], results exhibit non-linear behavior for MDV using textures. [Kha06a, Kha06b] deals with the isosurface and external 3D surface mapping techniques for MDV. The approaches used [Kha06a, Kha06b] in these paper do not utilize capabilities of modern GPU. Moreover texture based clipping [Kha06b] is based on clipping planes and do not discuss complex clipping geometries in context to

MDV. In this paper, we extend these ideas for MDV using hardware accelerated texture based techniques for the MDV. The texture-based techniques include volume rendering, clipping and isosurface extraction methods. Our results show similar non-linear trend. For performance analysis, we consider the 3D textures of charge density distributions in real material systems (MgO and MgSiO<sub>3</sub>, perovskite), which are investigated on routine basis by parallel quantum mechanical simulations [Cod05]. The datasets considered are of moderate size e.g. 256<sup>3</sup>, which is quite common for current scientific and engineering applications. This paper also presents the effect of shading, which has not been investigated previously in context of MDV.

Texture mapping [Cab94, Cul93, Wil94] is one of the widely used techniques for visualization of scalar data. Often clipping is used as an important tool with texture-based volume rendering for uncovering the hidden details. Clipping planes [Gel96] were previously used for clipping. Weiskopf et al. have proposed techniques for volume clipping with complex geometries, which are based on the depth structure and voxelization of the clip geometry [Wei03]. Some studies have used volume clipping based on stencil test [Wes98], isosurface clipping [For92] and interactive clipping combined with dual-resolution texture-based volume rendering [Kha05].

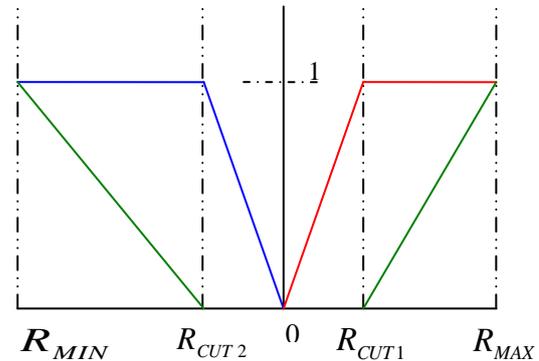
An isosurface represents a 3D surface at a constant scalar value within a volume. Lot of work for isosurface extraction [Lor87, Cli88, Sch92, Par98, Sch04, Kip05, Geo06] has been done over last two decades. Westermann [Wes98] proposed 3D texture based technique for isosurface extraction. GPU based raycasting [Had05, Kip05] approach has been effectively used for isosurface extraction. A number of techniques for combining 3D texture volume rendering with lighting and shading have been proposed [Gel96, Hau96, Wes98, Wei03]. In [Gel96] the sum of precomputed ambient and reflected light is stored in texture volume and standard 3D texture rendering is performed. In [Hau96] the voxel gradient and volume density are stored together as 3D texture. Due to recent advances of commodity graphics hardware, texture-based rendering is able to achieve acceptable frame-rates with high image quality [Wil94; Wei00; Cul93, Wil02, Wei03].

### 3. TEXTURE-BASED MDV TECHNIQUES

#### Transfer Function

Visualization of the scalar volume data requires transfer function for mapping the scalar value to the RGB color space. This section describes the transfer function (Figure 2) used in this paper except for the isosurface extraction where it is not needed, since we

are extracting the isosurface corresponding to single iso-value at a given instant. Here  $R$  is the scalar value. By changing  $R_{CUT1}$  and  $R_{CUT2}$ , we can highlight different region in the dataset.

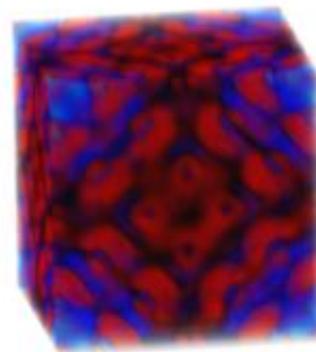


**Figure 2: Transfer Function for the Scalar Volume Data. Red, Green and Blue curve represents the transfer function  $T_R$ ,  $T_G$  and  $T_B$  respectively.  $R_{MIN}$  and  $R_{MAX}$  are the maximum and minimum value in the dataset.**

We have used linear functions for RGB values, which can be changed to exponential, parabolic depending upon the dataset and the user requirements.

#### Volume Rendering

The texture mapping approach utilizes 3D textured data slices [Cab94; Cul93; Wil94]. The 3D texture approach can sample the data in the s, t, or r directions freely so the slices are always oriented perpendicular to the viewer's line of sight. Tri-linear interpolation happens at hardware level thus allowing using arbitrary number of slices with an appropriate re-sampling on the slices. Figure 3 shows the volume rendering using 3D textures with alpha blending enabled.



**Figure 3: Volume Rendering with Alpha Blending**

MDV using 3D textures involves following steps. First step involves loading the dataset. Next step is to choose the number of slices perpendicular to the viewing direction for each texture. Third step is to use texture coordinate generation to texture the slice

properly with respect to each 3D texture data. In the fourth step, rendering of the texture slices is performed from back to front, towards the viewing position, with appropriate blending performed at each slice. Finally, next dataset is loaded and all of the above steps are repeated for this dataset. For clipping, isosurface extraction and volume shading, the calculations are performed in the GPU before fragments are finally rendered and after the texture and vertex coordinates are bound together in step 4. As the viewpoint and viewing direction is changed, data slice positions need to be recomputed and texture transformation matrix needs to be updated.

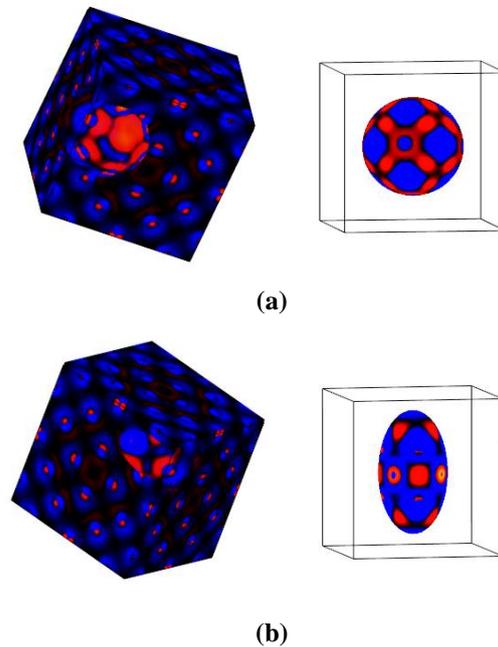
## Clipping

The purpose of clipping is to explore the hidden details in the dataset. The clipping can be either, volume probing, where fragments lying inside the clipped geometry are kept or volume clipping, where fragments lying outside the clipping geometry are kept. In this section we present clipping based on surface texture rendering (planar and box clipping) [Kha06b] and voxelized clipping approaches which can be interactively used with MDV.

Planar/Box clipping technique utilizes the surface rendering, which involves mapping texture only the visible surface area. Volumetric data is represented by a set of triangles on which the 3D texture is mapped. Initially six planar surfaces of simulation box are rendered. In this approach, we find single or multiples surfaces cutting the volume and then bound the intersecting surfaces in the form of simple polygons. These polygons determine the new set of externally visible surfaces of the volume and the textured data is mapped only on these externally visible polygons. Thus, only the surfaces defined by a set of visible clipping polygons (single or multiple) are rendered. For planar clipping, the intersection of plane with 12 triangles is considered Implementation details of planar and box clipping can be found in [Kha06b].

Voxelized clipping technique [Wei03] utilizes two textures, volume object and clipping object. The 3D texture of the clipping object uses binary representation. The alpha value of the clipping texture is set to 0 or 1 depending upon whether the voxel lies inside the clipping geometry or not. Depending upon the alpha value of the clipping texture GPU assigns the alpha value to the volume texture. The fragments lying outside the clipping texture are assigned value zero. These fragments are discarded using the alpha test for volume probing. For volume clipping the fragments lying outside are assigned the value 1. The clip object can be moved around by mapping the texture coordinates of the clipping texture although changing the shape of the

clipping object requires the loading of the new 3D texture. Since the method utilizes the binary representation there is a visible transition between the voxels having value 0 and 1.



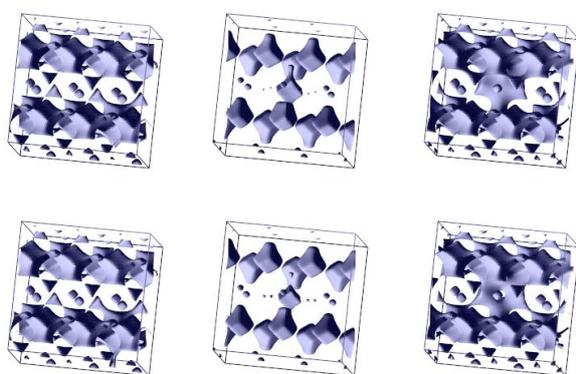
**Figure 4: Voxelized clipping. (a) Spherical Clipping Object. (b) Ellipsoidal Clipping Object**

To overcome this artifact induced by the use of the binary textures, we utilize the distance map similar to the one given by Lorenson [Lor93]. Evaluation of  $F(x, y, z)$  at any point produces a value  $< 0, = 0,$  or  $> 0$ , where  $F(x, y, z)$  gives the signed Euclidean distance of the voxel from the surface of the clipping object. Points that lie inside the clipping object are assigned negative values and points outside the clipping object are assigned the positive values. The points on the surface of the clipping object have a distance zero. If RGBA 3D textures are used, these distances get clamped to the range (0, 1). Floating-point textures retain the floating-point value stored in them as it is, thus removing any round of error due to clamping in RGBA textures. For different clipping geometries there needs to be a different clipping 3D texture. The same 3D clipping object can be reused for the more than one number of dataset in case of MDV. A new 3D texture needs to be loaded if different clipping geometries are used with different datasets. Figure 4 shows the clipping based on the spherical and ellipsoid clipping object.

## Isosurface Extraction

This section presents an isosurface technique, which avoids an intermediate generation of polygonal geometry for the isosurface. Our approach is similar to one proposed by Westermann [Wes98]. Since we can exploit the trilinear interpolation of 3D textures,

isosurface (Figure 5) generation using 3D textures is thus equivalent to picking up those voxels through which isosurface passes. Each element in the 3D texture is assigned the scalar value as its alpha component. Then the modified alpha test:  $(R_{MAX}) > isoValue > (R_{MIN})$  is applied to the resulting 3D texture to find the voxels containing the isosurface.  $R_{MAX}$  and  $R_{MIN}$  give the maximum and minimum value respectively for the corresponding voxel, which is used to test whether the isosurface corresponding to the particular isovalue passes through the voxel or not. Modern programmable GPUs can be utilized to perform this task efficiently without the loss of performance. Since the values in the 3D texture are clamped to the region  $[0, 1]$ ; this approach might not pick up all the areas for the resulting isosurface or the isosurface may not be continuous.



**Figure 5: Isosurface Extraction of  $MgSiO_3$  showing Mg, Si and O vacancy from left to right with isovalue 0.021. Top layer shows the isosurface after the optimization and lower layer shows the isosurface before the optimization**

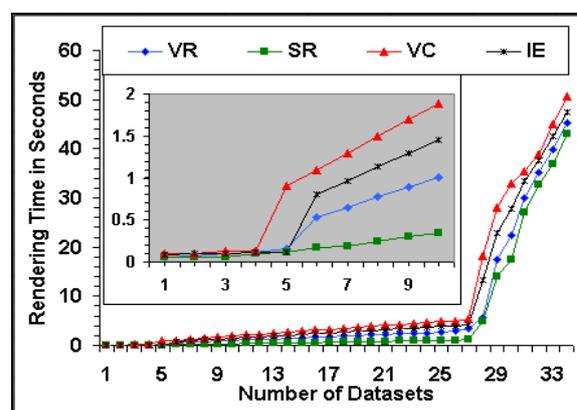
To overcome this problem we use the floating point 3D textures as described above. For isosurface generation we can use 16 or 32 bit floating point 3D textures. Most of the modern GPU's provide trilinear interpolation for 16 bit floating point 3D textures. This functionality is not available for the 32 bit floating point textures. Therefore, for using 32 bit floating point 3D textures for the isosurface extraction, trilinear interpolation needs to be implemented at the GPU level.

### Rendering Time for MDV Techniques

Our analysis involves the rendering time for volume rendering, clipping and isosurface extraction with and without volume shading in context of MDV. Figure 6 shows the rendering time as a function of number ( $N$ ) of datasets, which are visualized concurrently.

In case of volume rendering, alpha blending is enabled. For volume rendering, we get the frame rate of 15.6 fps for single dataset, which gradually fall with increasing number of datasets. Rendering time

starts to increase rapidly after 5 dataset. In case of box clipping using surface rendering, we get better frame rates as compared to the volume rendering time. With increasing number of datasets rendering time increases but still remains low as compared to regular volume rendering. Even for 10 datasets the frame rate is above 3, which is desirable for multiple datasets visualization. In case of voxelized clipping, we see similar behavior. The rendering time increases considerably after 4 datasets. After that we see the similar trend of increasing rendering time with increasing  $N$ . The voxelized clipping object can be manipulated in real time by changing the texture coordinates. Finally, for isosurface extraction the rendering time is pretty much similar to that for volume rendering. We can see a sudden jump in rendering time after 5 datasets and then there is a rapid increase in the rendering time. Isosurface can be changed in real time; since this technique does not generate the polygonal representation of the isosurface.



**Figure 6: Rendering time for various techniques without volume shading. The inset shows the first transition in the low  $N$  regime. (VR- Volume Rendering, SR – Surface Rendering, VC- Voxelized Clipping, IE- Isosurface Extraction)**

Considering the general trend in all the techniques Figure 6 without shading, we can see that rendering time shows two transitions; one around 5 dataset and another around 25 datasets. The first transition is due to the limited texture memory, which results in texture swapping. The second transition arises from the swapping between the main memory and the virtual memory. Since the second transition is due to the virtual memory, the rendering time increases much more rapidly after the second transition as compared to the increase after the first transition. In case of box clipping, such a high frame rate can be attributed to the fact that the number of slices being mapped is considerably less than the number of slices mapped in the case of actual volume rendering. With voxelized clipping approach, the frame rates are

comparatively smaller due to the extra textures required for clipping. The first transition in rendering time occurs little early as compared to other techniques because of the extra texture.

#### 4. VOLUME SHADING

Here we first give the brief description about the shading model we implemented and then do the performance analysis of the techniques mentioned in section 3 (volume rendering, clipping and isosurface extraction) when shading is enabled.

##### Shading Model

Illuminating the region of the interest in the data further facilitates the understanding. Volume shading adds the illumination term to the volume rendering. We have utilized the phong model of illumination for the clipped surface and isosurfaces. According to the phong model [Pho75], light at any given point is combination of diffuse, specular, and ambient component. These three components are added to determine the illumination or color of a point or polygon. Calculating the surface normal is done using the gradient. For each 3D texture a gradient texture corresponding to the scalar dataset is created which hold the normal for each voxel depending upon the scalar values. Therefore, for each 3D texture being visualized there is a gradient texture. Both these are passed to the GPU and lighting calculations are performed. This technique requires an additional texture per dataset, thus reducing the number of datasets that can be visualized without shading to half. Since we already have a pre-computed clipping texture, applying to the isosurface or the clipped volume is straightforward.

##### Rendering Time with Volume Shading

When shading is enabled, the maximum number of datasets that can be visualized together is reduced roughly to half. This is due to the requirement of one gradient texture per dataset. Gradient calculation can also be done but it would require 6 extra texture fetches for each voxel. Figure 7 shows the rendering time for the above techniques when the shading is enabled. Although decrease of rendering time with shading is primarily due to extra gradient texture, it is also related to the lighting calculation done by GPU for every visible fragment. The effect of shading can be clearly seen for all the four cases as the number of datasets for the transitions decreases and rendering time increases (Figure 7). The first transition occurs after 2 datasets, since while we are displaying 2 textures with shading; we are effectively dealing with 4 textures in the memory. The second transition occurs at 13 dataset in contrast to 27 datasets when shading is disabled. The frame rates are also affected

in a similar way. For 4 datasets, volume rendering frame rate is around 9 fps with shading while it is 0.67 fps with shading. Similar behavior can be seen when MDV for 9 or 16 datasets is done. For 9 datasets with voxelized clipping the frame rates are 0.47 and 0.10 with and without shading respectively. Since the maximum number of datasets that can be handled together is only 17.

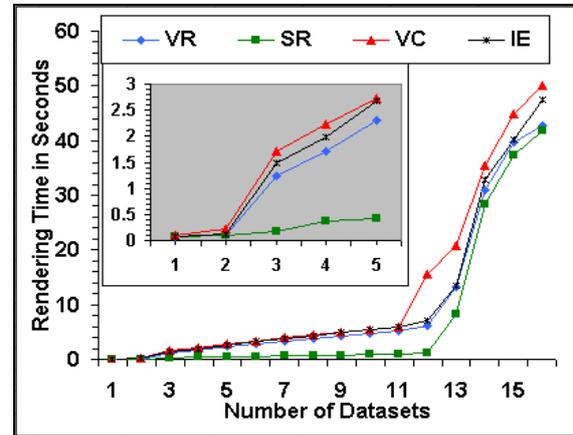


Figure 7: Rendering time for various techniques with shading. The inset shows the first transition in the low  $N$  regime. (VR- Volume Rendering, SR – Surface Rendering, VC- Voxelized Clipping, IE- Isosurface Extraction)

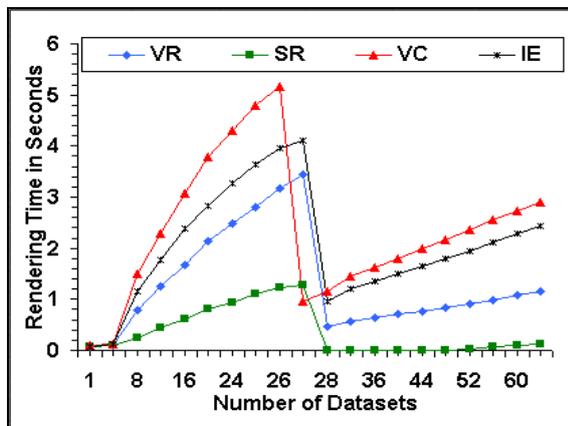
#### 5. DYNAMIC RESOLUTION (DR) FOR IMPROVED MDV

All the above texture based techniques (with and without shading) show the general trend that rendering time increases non-linearly as number of datasets ( $N$ ) increase. The average slope of the  $time-N$  curve in the  $high-N$  region tends to be larger than that in the  $low-N$  region. An improved MDV method is expected to reduce the time in overall, maintain a linear  $time-N$  relationship over the entire  $N$ -regime, and increase  $N_{MAX}$ . The dynamic resolution approach is similar to the one proposed in [Kha06a]. The dynamic-resolution (DR) method, switches from a high- to a low-resolution mode as  $N$  increases. The basis for texture generation at a reduced resolution is the fact that once the dataset becomes too large, individual texels tends to be sub-pixel. This is also true in the case of MDV since with increasing  $N$ , individual viewports corresponding to individual datasets become smaller for a given window display size. Our dynamic-resolution (DR) approach sub-samples the volume data for a reduced resolution according to some criteria. For sub-sampling we use the octree data structure which sub samples the original volume in to eight sub volumes. Dynamic resolution checks for the user-defined threshold limits to decide whether the resolution needs to be reduced

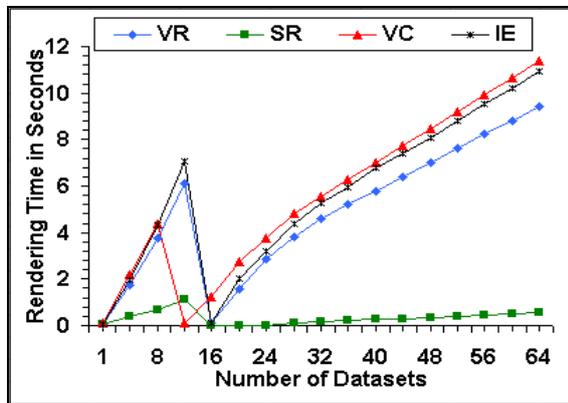
or not, every time a new texture is added. If increasing  $N$  does not break the threshold limit, more textures can be added at the current resolution. If the threshold limit is crossed, then the resolution of all the textures including the new ones is decreased. Reducing the resolution, frees up large amount of memory and thus more datasets can be visualized subsequently. Note that once the resolution is lowered, if the higher resolution is needed, the corresponding data have to be loaded again.

### Improved MDV Results Using DR

Using the above-proposed DR approach, threshold criteria is defined to maintain the linear behavior of rendering time with increasing number of datasets ( $N$ ).



(a)



(b)

**Figure 8: (a) Rendering time for various techniques without shading with DR. (b) Rendering time for various techniques with shading with DR. (VR- Volume Rendering, SR – Surface Rendering, VC- Voxelized Clipping, IE- Isosurface Extraction)**

The criteria for switching to lower resolution can be user defined. It can be based in rendering time, level of interactivity (number of frame rates) desired by the user, number of datasets or any other user defined criteria. The criterion avoids the relatively rapid

increase in the rendering times by examining the *time-N* slope. For instance,

$$\text{when, } R = \frac{T_{N+1} - T_N}{T_N - T_{N-1}} > 1.5, \text{ we switch to a lower}$$

resolution mode for the  $N+1$  and more datasets. Here  $R$ , defines the slope, and  $T_N$  defines the rendering time in seconds for the  $N$  number of datasets. The above criteria, reduces the resolution when the second transition occurs. The first transition does not satisfy the above criteria and is thus ignored. The condition could be modified to capture the first transition and reducing the resolution. When shading is disabled the resolution is decreased around 27 datasets (Figure 8a). After this point we see the linear behavior till 64 datasets. Similarly, when shading is enabled, transition to the lower resolution occurs around 12<sup>th</sup> dataset (Figure 8b). For voxelized clipping the transition to lower resolution occurs little early due to the use of extra texture. Reducing the resolution also frees up memory and thus allowing more number of datasets to be loaded. Using the different criteria linear *time-N* slope can be obtained for all the three regions of the curve shown in Figure 6 and Figure 7.

## 6. CONCLUSIONS

In this paper, we have presented various texture-based multiple datasets visualization (MDV) techniques. With volume rendering, clipping and isosurface extraction the trend is similar in all the cases. In all the cases, we see two transitions in rendering time. First is due to swapping between main memory and texture memory. After second transition swapping of the texture objects takes place between the main memory and texture memory which dominates the rendering time. When volume shading is enabled an extra texture is needed to pass the surface normal to the GPU which reduces the overall number of textures that can be visualized simultaneously. To improve the performance we exploit the Dynamic-Resolution approach. The DR reduces the storage space so that more datasets can be loaded and a linear *time-N* slope can be maintained.

## 7. REFERENCES

- [Cab94] Cabral, B., Cam, N. and Foran, J. Accelerated volume rendering and tomographic reconstruction using texture-mapping hardware. Proc. Symp. Volume Visualization. 91-98, 1994.
- [Chi97] Chi, Ed Huai-hsin, Konstan, J., Barry P, and Riedl, J. A spreadsheet approach to information visualization. Proc. of the 10th annual ACM symposium on User interface software and technology, pp. 79-80, 1997.

- [Cli88] Cline, H., Lorensen, W., Ludke S., Crawford, C., and Teter, B. Two algorithms for three dimensional reconstruction of tomographs, *Medical Physics*. Vol. 15, pp. 320-327, 1988.
- [Cod05] Codes. Electronic calculation packages: PWScf ([www.pwscf.org](http://www.pwscf.org)) and VASP ([cms.mpi.univie.ac.at/vasp](http://cms.mpi.univie.ac.at/vasp)), 2005.
- [Cru96] Crutcher, R.M., Baker, M.P., Baxter, H., Pixton, J. and Ravlin, H. *Astronomical Data Analysis Software and Systems V*, ASP Conference Series, Vol. 101., 1996.
- [Cul93] Cullip, T. J. and Newman, U. Accelerating volume reconstruction with 3D texture hardware. Technical Report TR93-027, University of North Carolina, Chapel Hill, N. C., 1993.
- [For92] Forgyson. Visual Kinematics Inc, (Mountain View, CA USA), FOCUS User Manual, Release 1.2., 1992.
- [Gel96] Van Gelder, A., and Kim, K. Direct volume rendering with shading via three-dimensional textures. *Proc. Symp. Volume Visualization*, pp. 23-30, 1996.
- [Hau96] Haubner, M., Krapichler, Ch., Lösch, Englmeier, K. H and Van Eimeren W. *Virtual Reality in Medicine – Computer Graphics and Interaction Techniques*. IEEE Transactions on Information Technology and Medicine, Vol 1, pp 61-72, 1996.
- [Geo06] Georgii J, Westermann R. A Generic and Scalable Pipeline for GPU Tetrahedral Grid Rendering, *IEEE Transactions on Visualization and Computer Graphics*, Vol.12, No.5, Sept./Oct. 2006.
- [Had05] Hadwiger M., Sigg C., Scharsach H., Buhler K. and Gross M., Real-time ray-casting and advanced shading of discrete isosurfaces. *Eurographics*, 2005.
- [Jan00] Jankun-Kelly, T. J., Ma, K. A spreadsheet interface for visualization exploration *IEEE Proc. of the conference on Visualization*, 2000.
- [Kar06] Karki, B B., and Khanduja G. *Computer Simulation and Visualization of Vacancy Defects in MgSiO<sub>3</sub> perovskite*. MSMSE, 2006.
- [Kha05] Khanduja, G. and Karki, B. B. Visualization of 3D scientific datasets based on interactive clipping. WSCG, ISBN 80-903100-9-5., 2005.
- [Kha06a] Khanduja, G. and Karki, B. B. Multiple Dataset Visualization using Isosurface extraction. *VIIP*, pp. 541-547, 2006.
- [Kha06b] Khanduja, G. and Karki, B. B. A Systematic Approach to the Multiple Dataset Visualization of Scalar Volume Data. *GRAPP*, pp. 59-66, 2006.
- [Kip05] Kipfer P. and Westermann R. GPU construction and transparent rendering of isosurfaces. *Proceedings of Vision, Modeling and Visualization*, 2005.
- [Lev94] Levoy, M. Spreadsheets for Images, *ACM Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, Vol 28, pp. 139—146, 1994.
- [Lor87] Lorensen, W.E. and Cline H.E. Marching Cubes: A high-resolution 3D surface reconstruction algorithm. *Computer Graphics, Proc. of SIGGRAPH*, 21, pp. 163-169, 1987.
- [Lor93] Lorensen, W. E. Geometric clipping using Boolean textures *Visualization, Proc., IEEE*, 1993.
- [Mei00] Meibner, M., Huang, J., Bartz, D., Mueller, K. and Crawfis, R. A practical evaluation of popular volume rendering algorithms, *Proc. IEEE Symp. Volume Visualization*, pp. 81-90, 2000.
- [Par98] Parker S., Shirley, P., Livnat, Y. Hansen, C., and Sloan, P. Interactive ray tracing for isosurface rendering, *Proc. Visualization*, pp. 233-238., 1998.
- [Pho75] Phong, Bui-Tuong. Illumination for Computer-Generated Pictures, *Communications of the ACM*, vol. 18, no. 3, pp. 311-317, 1975.
- [Sch04] Schulze, J.P., and Forsberg, A.S. User-friendly volume data set exploration in the Cave, <http://www.calit2.net/~jschulze/publications/Schulze2004b.pdf>., 2004.
- [Sch92] Schroeder, W., Zarge, J. A., and Lorensen, W.E. Decimation of triangle meshes, *Computer Graphics. SIGGRAPH Conf. Proc.*, pp. 65-70, 1992.
- [Wei00] Weiler, M., Westermann, R., Hansen, C., Zimmerman, K., and Ertl, T. Level-of-detail volume rendering via 3D textures. *Proceedings of IEEE Volume Visualization*, pp 7-13, 2000.
- [Wei03] Weiskopf D., Engel, K., and Ertl, T. Interactive Clipping Techniques for Texture-Based Volume Visualization and Volume Shading. *IEEE Transactions on Visualization and Computer Graphic*, 9, 298-312, 2003.
- [Wes98] Westermann, R. and Ertl, T. Efficiently using graphics hardware in Volume Rendering Applications. *SIGGRAPH Conf. Proc.*, pp. 169-179, 1998.
- [Wil02] Wilson B., Ma, K. and McCormick, P. S. A hardware-assisted hybrid rendering technique for interactive volume visualization. *Proc. of Volume Visualization and Graphics Symposium*, Vol 9, Issue 2, pp 123-130, 2002.
- [Wil94] Wilson, O., Van Gelder, A. and Wilhems, J. Direct volume rendering via 3d textures. Technical Report UCSC-CRL-94-19, Jack Baskin School of Eng., Univ of California at Santa Cruz., 1994.

# Physically Realistic Interface for a user inside VR

Masahide Hashimoto  
Hosei University  
Koganei Kajinocho 7-3-2  
184-8584, Tokyo, Japan

mhashimoto@philos.k.hosei.ac.jp

Kenji Miyamoto  
Hosei University  
Koganei Kajinocho 7-3-2  
184-8584, Tokyo, Japan

miyaken@hosei.ac.jp

## ABSTRACT

Interaction between a user inside a virtual space and virtual objects is described. Incorporating a user's live image into a virtual space enables him/her to exist in the same place as objects and to interact with them. There are, however, the lack of guideline to design interaction style makes users confused about how to use it. This article proposes a common implementation framework that forces the interaction to be physically realistic. As a result, the look & feel of interaction becomes uniform. In a ball game application and virtual electronic controllers made by this framework, a user manipulated/played in the same way as those in the real world. This framework could provide novice-friendly or hands-free interface ubiquitous around public or daily places.

## Keywords

Keywords are virtual reality, bodily interaction and interactive animation.

## 1. INTRODUCTION

Interaction between a user inside a virtual space and virtual objects is described.

Incorporating a user's live image into a virtual space enables him/her to exist in the same place as objects and to interact with them. Stepping-in presentation [Yos03a] is a presentation system in which a user can use his/her live image not only to give bodily performance but also to do operation such as page changing. [Kan04a] proposed a communication system in which participant's live image's can bodily express their emotion by touching each other. EYETOY [Son03a] is an interactive game that uses user's live image. Mirror interface [Hos03a] is an interface that enables a user's live image to manipulate a remote object.

There are, however, no guidelines to design interaction style inside a virtual space. For example, [Yos03a] and [Hos03a] use a specific color to manipulate as a switch switches red/blue (ON/OFF), [Kan04a] uses overlapping user's area, and [Son03a] uses whether the user is moving or not as a switch switches waving in a specific place. As a result, a user does not how to manipulate it.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Copyright UNION Agency – Science Press, Plzen, Czech Republic.

In order to provide in-VR interaction of uniform look & feel and its common implementation platform, this article proposes a framework that forces the interaction to be physically realistic.

In this framework, an allowed interaction consists of a user's action in a virtual space and a reaction of each virtual object that has to be interpreted naturally as operation in the real world. Note that user's actions are, by definition, limited to be those performed to some virtual object. This interaction is described by a rule that associates an action such as touching, detouching or moving with a reaction such as rotation or transformation. Using this framework, a developer can concentrate to working on an object's reaction without being bothered by choosing interaction style.

To demonstrate the effectiveness of this framework, a virtual object such as dial, a toggle switch, a spring, a pickable object, and a bouncing object were implemented. More over, using these objects, two application products were also implemented. V-VolleyBall is a ball game application made of a bouncing object. In this game, the players see themselves in the screen playing as if with a real ball. V-Rimokon is a virtual electronic controller that controls consumer electronic at a remote place made of a dial object. This controller used was able to in a same way as manipulating a physical dial. Experimental use showed that all the users could manipulate/play at sight.

Interactions build in this framework could be novice-friendly and hands-free interface ubiquitous around public or daily places.

## 2. Virtual Manipulation

In *Virtual Manipulation (VM)*, a user manipulates virtual objects watching him/herself in a virtual space. Figure 1 shows a scene in which a user is manipulating a dial in a virtual space. In this figure, the hand outside the screen is in the real world, whereas the user manipulates the dial watching the hand image inside the screen.

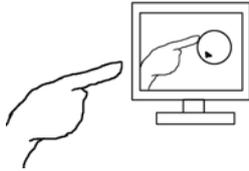


Figure 1. Using VM

VM consists of a *virtual interface*  $I_V$ , a *real interface*  $I_R$  and an *interpretation*  $J$  of  $I_V$  in  $I_R$ . This section, after giving the definition of VM, presents two components of  $I_V$ , *events* and *reactions*, the former of which describes basic actions of a user in the virtual space, and the latter describes responses of each virtual object.

### 2.1 Definition of VM

#### 1. Basic Definition

The virtual interface  $I_V$  is defined by a triple  $\langle E_V, A_V, \eta \rangle$ , where  $E_V$  is a set of events in a virtual space,  $A_V$  is a set of reactions of objects in the virtual space and  $\eta : E_V \rightarrow A_V$  is virtual response a function that returns a reaction given an event in the virtual space. Here, a reaction  $a \in A_V$  is a function that changes the state of given objects. If the object under consideration is referred to as the *target* of  $I_V = \langle E_V, A_V, \eta \rangle$ .

The real interface  $I_R$  is defined by a triple  $\langle E_R, A_R, \xi \rangle$ , where  $E_R$  is a set of events in the real world,  $A_R$  is a set of reactions of objects in the real world and  $\xi : E_R \rightarrow A_R$  is real response a function that returns a reaction given an event in the real world.

An event  $e \in E_V$  is interpreted as an event  $e' \in E_R$  by *event interpretation*  $\phi : E_V \rightarrow E_R$  and a reaction  $a \in A_V$  is interpreted in a reaction  $a' \in A_R$  *reaction interpretation*  $\psi : A_V \rightarrow A_R$ .

By letting  $J$  be a pair  $\langle \phi, \psi \rangle$  of interpretations above, a VM can be formulated as a triple  $\langle I_R, I_V, J \rangle$ .

Note that  $E_R, E_V$  and  $\phi$  are common to objects and, on the other hand,  $A_R, A_V, \eta, \xi$  and  $\psi$  are determined by each object.

#### Example

In the example shown in Fig. 1,  $E_R$  is the set that consists of rubbing in the real world.  $E_V$  is the set that consists of movements of the point at which a user

contacts with the object.  $\phi$ , given a movement of a contact point in a virtual space, returns rubbing in the real world.

In Fig. 1, the object is a dial, which determines the followings:  $A_R$  is a rotation of the dial in the real world.  $A_V$  is a rotation of the figure of the dial in the virtual space.  $\eta$ , given a movement of a contact point, returns a rotation of the dial figure in the virtual space.  $\xi$ , given rubbing the dial in the real world, returns a rotation of the dial in the real world.  $\psi$ , given a rotation of the dial figure in the virtual space, returns a corresponding rotation of the dial in the real world.

#### 2. Reality Condition

VM must satisfy the following requirements (*Reality Condition, RC*) in order for the interaction to be physically realistic.

1. User's movement in the real world and movement in the virtual space are simultaneous and similar.
2. The reaction  $a \in A_R$  that is caused, through  $\xi$ , by the event  $e' \in E_R$  represented, through  $\phi$ , by the event  $e \in E_V$  must be equal to the reaction  $a' \in A_R$  that is represented, through  $\psi$ , by the reaction  $a \in A_V$  caused, through  $\eta$ , by event  $e \in E_V$ . That is  $\xi \cdot \phi = \eta \cdot \psi$  holds. Figure 2 shows the commuting diagram for this equality.

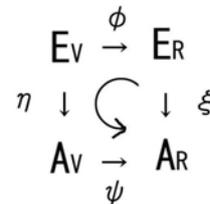


Figure 2. Diagram commutes.

In the following, VM request RC to hold.

#### Example (RC for a dial)

If user's live date taken by a video camera, a data glove, etc is used to incorporate the user, the user in the real world and the user in a virtual space moves simultaneous and similarly.

The rotation in the real world that is caused, through  $\xi$ , by the rubbing in the real world represented, through  $\phi$ , by the movement of a contact point in the virtual space must be equal to the rotation in the real world that is represented, through  $\psi$ , by the rotation in the virtual space caused, through  $\eta$ , by the movement of a contact point in the virtual space.

## 2.2 VM for two-dimensional virtual space

### Events

For the case that the user in the virtual space is a two-dimensional image, we propose following events as  $E_V$ .

Event interpretation  $\phi$  is shown in Table 1 and the attributes for each event are shown in Table 2.

In the following and in the tables, the area of user's two-dimensional image is denoted by  $U$ , a two-dimensional figure of the object figure is denoted by  $F$  and a center of gravity for a two-dimensional figure is an average of the position of the point inside the figure.  $G(U,F)$  given  $U$  and  $F$ , returns a center of gravity for the area where  $U$  and  $F$  overlap.

### Reactions for a dial

$F, A_V, \eta$  and  $\phi$  for a dial are defined as follows.

$F$ : A circle with a marker indicating a rotation angle (Fig 3).

$\phi$ : The mapping which maps a rotation of a dial figure in the virtual space to the same rotation of a dial in the real world.

$\xi, \eta$ : Shown in Table 3.



Figure 3. The dial figure in the virtual space

Note that the dial satisfies RC. We will see this in detail later at Physical Reality of the dial example in section 3.

## 3. Objects virtual interface in VM

An object with virtual interface can easily be described in VM. In this section, a dial, a toggle switch, a spring, a pickable object, and a bouncing object are described as examples of such object. In the explanation of each object, first, a specification of a real object is shown, next, actions are presented, and finally, RC is checked. The actual implementation of each object is shown in the next section.

### 3.1 A dial

#### Physical specification

A physical dial (Fig. 4) can be rotated by rubbing its circumference (after touching it).



Figure 4. A dial

#### Reactions

The target  $F$  is a circle with a sign indicating its rotation angle (Fig. 5).  $F$  becomes opaque on the event **Overlap**, becomes translucent on the event

$\phi(e) \in E_R$	$e \in E_V$
A user <b>touches</b> an object	Overlap : $U$ overlaps $F$
A user <b>releases</b> an object	Release : $U$ separates $F$
A user <b>moves</b> for <b>touching</b> an object	Move : $U$ moves for overlapping $F$

Table 1. Event interpretation  $\phi$  for two-dimensional virtual space

Event	Attributes
Overlap	$x$ : an abscissa of $G(U,F)$
	$y$ : an ordinate of $G(U,F)$
Release	
Move	before_x : an abscissa of $G(U,F)$ before movement
	before_y : an ordinate of $G(U,F)$ before movement
	after_x : an abscissa of $G(U,F)$ after movement
	after_y : an ordinate of $G(U,F)$ after movement

Table 2. Attributes of each event in two-dimensional virtual space

**Release**, and rotates on the event **Move**.



Figure 5. Dial figure

$\xi, \eta$  : Shown in Table 3.

$\phi$  : Rotation of  $F$  means the same rotation of the dial in the real world.

*Physical Reality*

If overlapping area moves near the circumference of the target  $F$ , its moving speed is equal to that the sign, which is near the circumference.

**3.2 A toggle switch**

*Physical specification*

$e \in E_R$	$\xi(e)$	$\phi(e)$	$\eta \cdot (\phi(e))$
<b>touches</b> a dial	Give feeling of touch	Overlap	$F$ becomes opaque
<b>releases</b> a dial	Give feeling of release	Release	$F$ becomes translucent
<b>rubs while touching</b> a dial	Rotates length rubbed	Move	$F$ rotates by $\arctan\left(\frac{\text{Move.after}_y - F.y}{\text{Move.after}_x - F.x}\right) - \arctan\left(\frac{\text{Move.before}_y - F.y}{\text{Move.before}_x - F.x}\right)$ degrees.

Table 3.  $\xi$  and  $\eta$  of a dial

$e \in E_R$	$\xi(e)$	$\phi(e)$	$\eta \cdot (\phi(e))$
<b>moves downward</b> ON to OFF	ON state switches to OFF	Move	If $\text{Move.before}_y > C_y$ and $\text{Move.after}_y < C_y$ , $F$ becomes state of OFF (Fig.7. Right side) .
<b>moves upward</b> OFF to ON	OFF state switches to ON		If $\text{Move.before}_y < C_y$ and $\text{Move.after}_y > C_y$ , $F$ becomes state of ON (Fig.7 Left side)

Table 4.  $\xi$  and  $\eta$  of a toggle switch(  $C_y$  is an ordinate of the center of  $F$ .)

$e \in E_R$	$\xi(e)$	$\phi(e)$	$\eta \cdot (\phi(e))$
<b>touches</b> a spring	A spring expands and contracts	Overlap	Position of $F$ becomes Overlap.y
<b>expand and contract</b> a spring while <b>touching</b>	A spring expands and contracts	Move	Position of $F$ becomes Move.after_y
<b>releases</b> a spring after pressing	A spring oscillates	Release	If $\text{Move.after}_y$ is not equal 0, $F$ oscillates.

Table 5.  $\xi$  and  $\eta$  of a spring

A Toggle switch (Fig. X) has the states up and down. To change the state up to down (ON→OFF), slant the bar downward, and vice versa.



Figure 6. A toggle switch

*Reactions*

The target  $F$  of the virtual toggle switch has two states ON and OFF as shown in Fig. X. The event **Move** which sweeps  $F$  upwards / downwards switches these states.

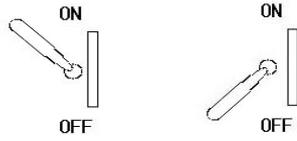


Figure 7.F/ON and F/OFF

$\phi$ : The switching ON to OFF or OFF to ON in the virtual space means the switching of the toggle switch in the real world.

$\xi, \eta$ : Shown in Table 4.

#### Physical Reality

Two states (ON,OFF) can be switched by  $U$ , by the same action as the toggle switch in the real world. That is, if the user's image moves upward overlapping F/OFF, the figure turns to F/ON and vice versa.

### 3.3 A spring

#### Physical specification

As the finger moves down pressing the spring (Fig. 8), it expands and contracts. The spring oscillates if the finger is released after pressing it.



Figure 8.A spring

#### Reactions

The target  $F$  is a figure shown Fig. 8.  $F$  expands and

$e \in E_R$	$\xi(e)$	$\phi(e)$	$\eta \cdot (\phi(e))$
Two fingers <b>pick</b> an object up	Give feeling of touch	$\{\text{Overlap.x,Overlap.y}\} \notin U$	$F$ becomes opaque
<b>release</b> an object	Give feeling of release	Release	$F$ becomes translucent
<b>Touch</b> with one finger	Give feeling of release	$\{\text{Overlap.x,Overlap.y}\} \in U$	$F$ becomes translucent

Table 6.  $\xi$  and  $\eta$  of a pickable object.

contracts by the event **Move**, and oscillates by the event **Release** for contraction.

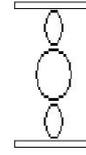


Figure 8. A spring figure

$\xi, \eta$ : Shown in Table 5.

$\phi$ : The size of expanding, contracting and oscillation of  $F$  means those the spring in the real world.

#### Physical Reality

The size of expanding and contracting of the target  $F$  is determined by the position where it is overlapped by  $U$  and the oscillation is determined by the size contracted as a spring with some elasticity in the real world.

### 3.4 A pickable object

#### Physical specification

A pickable object can be picked if a user picks it up with two fingers and dropped if the user releases it. Picking and dropping can be judged from a sense of touch.

#### Reactions

The target  $F$  is a figure shown Fig. 9.  $F$  becomes opaque by the event **Overlap** which is caused by two separate overlapping areas, and becomes translucent by **Overlap** and the event **Release**.



Figure 9. A pickable object figure

$\xi, \eta$  : Shown in Table 6.

$\phi$  : Becoming opaque and translucent in the virtual space means notifying touching and releasing some object in the real world.

*Physical Reality*

The situation that the gravity center of the overlapped area is outside  $U$  (Fig. 10) is usually the case that the object is touched at two points on mutually opposite side of the object.



Figure 10. Point(Overlap.x,Overlap.y) for picking

### 3.5 A bouncing object

*Physical specification*

If the moving object contacts with a user in the real world, the object bounces by elastic collision.

*Reactions*

The target  $F$  is a figure shown Fig. 11. The traveling direction of  $F$  is changed by the event **Overlap**.

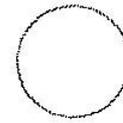


Figure 11. A bouncing object figure

$\xi, \eta$  : Shown in Table 7.

$\phi$  : The bouncing angle of  $F$  means the bouncing angle of the object in the real world.

*Physical Reality*

Velocity change caused by the contact with  $U$  is similar to that of a real ball.

## 4. Real System

The method of the user's clipping is the back ground subtraction.

### 4.1 A dial

The tip of a finger is moved toward the circle until it overlaps (Fig. 12. up side) and is moved along the circumference of the circle (Fig. 12. down side).

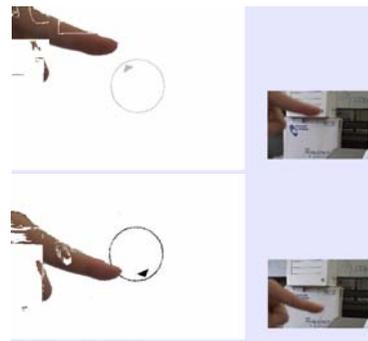


Figure 12. A dial

### 4.2 A switch

Touching the top of the switch by the tip of a finger (Fig. 13. up side) and the tip of a finger is moved below (Fig. 13. down side).

$e \in E_R$	$\xi(e)$	$\phi(e)$	$\eta \cdot (\phi(e))$
<b>touches an object</b>	An object bounces	Overlap	$\vec{n} = \frac{\begin{pmatrix} F.x - \text{Overlap}.x \\ F.y - \text{Overlap}.y \end{pmatrix}}{\sqrt{(F.x - \text{Overlap}.x)^2 + (F.y - \text{Overlap}.y)^2}}$ $\vec{v}' = \vec{v} - 2\vec{n}(\vec{v} \cdot \vec{n})$ <p><math>\vec{v}</math> is the direction of travel of F</p> <p><math>\vec{v}'</math> is the direction of travel of F after bouncing</p>

Table 7.  $\xi$  and  $\eta$  of a bouncing object

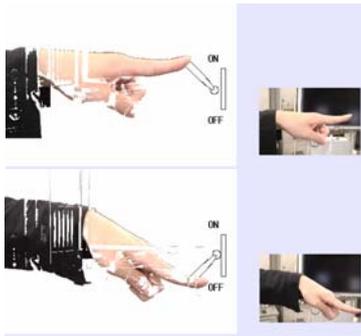


Figure 13. A switch

### 4.3 A spring

The tip of the finger is moved toward the spring figure until they overlap (Fig. 14. up side), the tip of the finger is moved below (Fig. 14. middle side) and the spring figure begins to oscillating when the tip of the finger is separated from the spring figure (Fig. 14. down side).

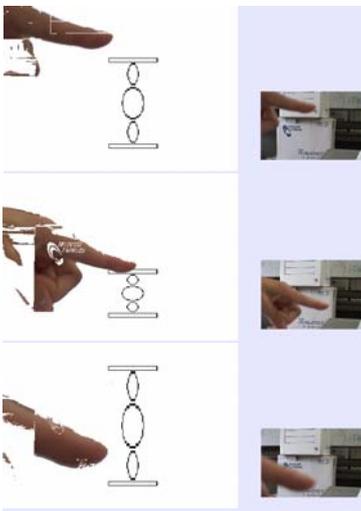


Figure 14. A spring

### 4.4 A pickable object

The two tip of the fingers are moved toward the circle until they overlap (Fig. 15. up side) and the circle is picked up by them (Fig. 15. down side).

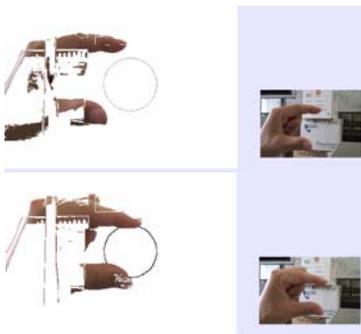


Figure 15. A pickable object

### 4.5 A bouncing object

The hand contacts with the moving circle (Fig. 16. up side) and the moving circle bounces (Fig. 16. down side).

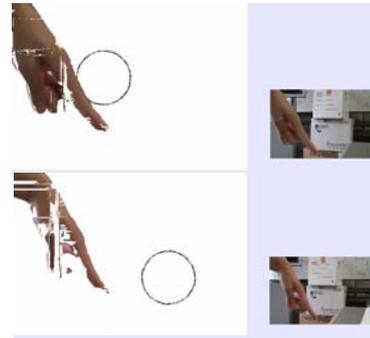


Figure 16. A bouncing object

## 5. Application

This section presents some screenshots of the scene in the game application (V-VolleyBall) and a remote controller (V-Rimokon) both of which use objects described in section 3.

### 5.1 V-VolleyBall

V-VolleyBall is a volleyball game application (Fig. 17. left side). Two players can play volleyball watching a virtual court in which the two players play with a virtual ball (Fig. 17. right side), a bouncing object described in section 3.



Figure 17. Using V-VolleyBall

### 5.2 V-Rimokon

V-Rimokon is intended to be used as a remote controller for the consumer electronic. The user can change the volume on the television in the same room but a little apart from him (Fig. 18. left side) by turning the virtual dial (Fig. 18. right side) described in section 3 on the screen



Figure 18. Using V-Rimokon

## 6. Result

### 6.1 Evaluation of Objects and V-VolleyBall

Usability experiment is done on low-school students (age 14), high-school (age 17) and graduate-students (age 23), each about ten subjects. In all groups, it is found that user can manipulate/play at sight without any instruction.

### 6.2 Developer's process

First, an object's figure is decided. Next, the action is chosen from among the fixed set  $E_V$ . Finally, the object's reaction is described.

## 7. Discussion

A framework for designing interaction between a user inside a virtual space and virtual objects is presented. In this framework, the granularity of description of events and reactions in virtual world is the same as that of description of those in the real world and each event-reaction rules in the both worlds are imposed to be semantically equivalent. As the result, interactions in this framework can easily be designed from given physical specification about real object and is necessarily physically realistic. The resulting interaction is physically realistic in the sense that not only its look and feel but also the needed actions to interact with objects are the same as those in the real world. V-VolleyBall may seem difficult to play because of the mismatch between the two places, the one where on the screen the user is watching the virtual ball running against to his virtual hand and the one where his actual hand volleying in the real world. Contrary to this expectation, however, experiment shows that the above mismatch does not matter in playing game.

This framework can also be applied to a three dimensional virtual space using pose-sensing device, for example, the data glove. Interactions build in this framework can be used as novice-friendly or deviceless interface ubiquitous around public or daily places.

### Limitation

In VM, although the user's image is touching an object in the virtual space, his real body is touching nothing. Therefore, an actual user doesn't have the sense of touch. This situation makes the user's action in the virtual space unstable because the proper positioning is a little difficult without a physical surface. This problem can be solved by a haptic interface device that feeds back contact in a virtual

space to the sense of touch. However, the haptic device is applicable only when the point on the virtual body at which is to contact with objects is preliminarily fixed (as the tip of some finger, for example).

### Related Work

Other methods for manipulating virtual objects use a hand glove. [Dor01a] present a system that allows a user to manipulate the virtual object by recognizing his/her hand's shape and gesture. Although, it can easily and accurately be obtained by hand glove, it can not be used in daily life.

### Future Work

This interaction can be extended to 3D by the model-based reconstruction from 2D to 3D or encoding 3D-meaning in 2D-actions.

Deformation of virtual objects using a clayish object would be an interesting challenge along with this approach.

### Conclusion

A framework that forces reality was proposed.

Interactions implemented in this framework were able to be used easily at sight.

## 8. ACKNOWLEDGMENTS

The authors thank to Hiroshi Mitsumata, Akinori Yoshino, Kanehata Shinichi (graduate of Hosei Univ.) for the preliminary works.

## 9. REFERENCES

- [Dor01a] Dorfmueller-Ulhaas.K, Schmalstieg.D. Finger tracking for interaction in augmented environments.*Proc. IEEE and ACM International Symposium on*, pp.55-64, 2001.
- [Hos03a] Hosoya.E, Kitabata.M, Sato.H, Harada.I, Nojima.H, Mrisawa.F, Mutoh.S, and Onozawa.A. A Mirror Metaphor Interaction System: Touching Remote Real Objects in an Augmented Reality Environment.*ISMAR2003*, pp.350-351, 2003.
- [Kan04a] Kanehata.S, Yoshino.A, and Miyamoto.K. Virtually Touchable Video Communication. *The Third International Conference on Information*, pp.589-592, 2004.
- [Son03a] SONY. EYE TOY. <http://www.eyetoy.com.2003>
- [Yos03a] Yoshino.A, Mitsumata.H, and Miyamoto.K. Stepping-in Presentation. *Proc. International Workshop on Multimedia Technologies in E-Learning and Collaboration(WOMTEC)*, 2003.

# Virtual Environments and Human-Computer Interaction Anxiety: An Experimental Study

Nasser Nassiri  
Higher Colleges  
of Technology,  
Dubai Women's College,  
Dubai,  
United Arab Emirates  
nasser.nassiri@hct.ac.ae

Norman Powell  
The University of Manchester. Centre  
for Excellence in Enquiry-Based  
Learning  
(CEEBL),  
Manchester, UK  
norman.powell@manchester.ac.uk

David Moore  
Leeds Metropolitan  
University.  
Innovation North. Faculty of  
Information and  
Technology, Leeds, UK  
d.moore@leedsmet.ac.uk

## ABSTRACT

In a world where interaction is a significant part of everyday life, personal space (PS) – an invisible bubble surrounding human body - is important. This is because PS functions as a comfort zone during interaction, a factor in designing physical environments, and invading such space produces anxiety and discomfort. Little is known, however, about how personal space might operate in virtual environments and what features the interface model of these environments should incorporate in order to reduce the anxiety produced by the invasions of such space in these environments. To begin to address this, we have conducted two experiments concerning personal space invasions (PSI) in collaborative virtual environments (CVE). Results suggest that reactions to PSI in CVE tend to differ in various ways from reactions to PSI in the physical world though some participants experienced anxiety when their avatar personal space was invaded. Recommendations to incorporate some features in the interface model of these environments are presented.

## Keywords

Collaborative virtual environment, personal space invasion, anxiety level.

## 1. INTRODUCTION

Personal space (PS) in the physical world is an area with invisible boundaries surrounding individuals which functions as a comfort zone during interpersonal communication [Dos69][Aie87]. It is influenced by many factors such as age [Hay83], culture [Aie87], gender [Gif96], and environmental factors such as room size [Eva96], room ceiling height [Coh84a], and environment location (i.e. indoors or outdoors) [Coh84b]. Personal space is often referred to as “interpersonal distance” – the distance apart from each other that conversational partners adopt. Personal space invasion occurs

when an individual enters another's personal space and thus produces discomfort and anxiety. The concept of personal space invasion anxiety level (PSIAL) refers to the degree of anxiety generated from an invasion of someone's personal space.

The study of personal space is of crucial importance, as evidenced by the fact that designs in the physical world take account of personal space issues.

Wiles [Wil78] for example found a relationship between the personal space allocated and the anticipated time of the event; thus telephone booths are allocated small space because the time people spend in them is little whereas homes for the elderly and prisons are allocated much larger spaces. Further, interpersonal distance can influence the level of understanding between interactants. Latan [Lat95] found that the further individuals are away from the source of communication the less influence it has over them. In medical clinics, people are more forthcoming when there is about 5ft space between themselves and the therapist. Moreover, [som58] found that altering the layout of chairs from being in rows in a clinic to being in circles increases the level of interactions between the patients. Clearly, then, personal space is an important issue in people's

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright UNION Agency – Science Press, Plzen, Czech Republic.

lives in the physical world. As people start to spend more time in the virtual world, it is important to investigate whether personal space exists in these environments and if so, how designers of the virtual worlds should seek to cater for it.

The rest of this paper is organized as follows: section 2 presents the research goals of this work. Section 3 reports on an experiment that was conducted to investigate the influence on personal space of avatar gender. Section 4 reports on an experiment that was conducted to investigate the influence on personal space of the virtual environment layout. Section 5 discusses the obtained results. Section 6 provides a summary of the research re-iterates its conclusions, and makes suggestions for further work.

## 2. RESEARCH GOALS

There is some evidence to suggest that the concepts of personal space and PSIAL have indeed been transferred to collaborative virtual environments. For example, [Bai01] conducted an experimental study in immersive virtual environments (IVEs) and found that individuals avoided violating others' personal spaces. Sommer [Som02] and Krikorian et al [Kri00] found similar results. Becker and Mark [Bec98] found that people in a desktop collaborative virtual environment that is accessible from the internet get annoyed if another avatar comes too close. Jeffrey [Jef98] observed the same virtual environment as Becker and Mark for several weeks and noticed that individuals sometimes maintain a distance between their avatars when they are interacting and tend to show some discomfort feeling when other avatars invade their avatar personal spaces. Despite the above research results, however, much is still unknown concerning PSIA in CVEs, and the research outlined in this paper seeks to address some of these unknowns. For example, what are the reactions of the CVE users to PSI? Do CVE users keep a comfort zone when they are interacting? Does avatar gender affect the PSI reaction in the CVEs? Further, with regards to environment architecture, does the environment's layout affect the PSI reaction in the CVEs as it does in the physical world and if it does, how should CVE designers manage it in order to design highly effective CVEs? In order to answer these questions, "The Avatar Gender" and the "Environment Layout" experiments have been conducted by the authors in ActiveWorlds<sup>1</sup>, an internet based CVE, to investigate respectively the impact of avatar gender and the environment layouts on PSIAL in the CVEs

---

<sup>1</sup> [www.Activeworlds.com](http://www.Activeworlds.com)

## 3. THE AVATAR GENDER EXPERIMENT

In this experiment participants, of both genders, had their avatars' personal space "invaded" by another avatar of (either the same or the opposite gender), and reported their anxiety levels through the use of a post experiment questionnaire. The experiment was conducted to investigate the effect of gender on the PSIAL in the CVEs, as several studies have shown that gender has an impact on the personal space in the physical world [Bur98]. Such studies have suggested that (a) females interact at closer distances than male only groups [Aie71], (b) mixed-gender groups interact at closer distances than male only groups [Bax70], (c) females allow closer approaches from others than males allow [Pat87] and (d) female and mixed-gender interactants use touch more than male only interactants [Eli75]. Similarly, Hewitt and Henly [Hew87] identified an order for the four gender combinations of personal space invasion in the physical world: men allow women to invade their personal space to the highest degree, followed by women allowing other women to invade their personal space, then men allowing men to invade personal space, and finally women allowed men to invade their space the least.

The avatar gender experiment involved 40 participants, each of whom had their avatar's personal space "invaded" by the avatar of a further participant - a "confederate" - who was acting under instructions from the researchers. The invasions took place in an already built virtual environment using ActiveWorlds; see Figure 1 for a room in this environment.



**Figure 1: A room in the virtual environment where invasion occurred**

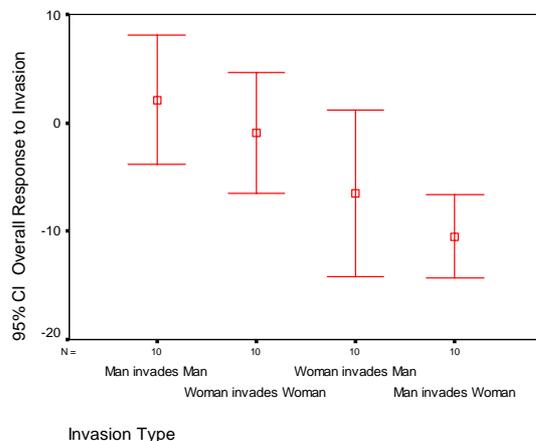
The results of the experiment suggested that the combination of the gender of the invading avatar and the avatar being invaded did have an influence on the personal space invasion anxiety level. The

descriptive statistics of anxiety with the means and 95% confidence intervals of each gender invasion group are shown in table 1 where, for example, 'M-m' indicates a male avatar invading another male avatar's space, and 'M-w' indicates a male avatar invading a female avatar's space. In this table, the anxiety means of all the groups are either negative or near the value 0 - an indication that the participants in general did not register anxiety when invaded, and indeed tended to be positive about it.

	N	Mean	Std. Dev	Std. Error	95% Confidence Interval for Mean		Min	Max
					Lower Bound	Upper Bound		
M-m	10	2.10	8.36	2.64	-3.88	8.08	-11	17
W-w	10	-.90	7.81	2.47	-6.49	4.69	-14	13
W-m	10	-6.50	10.80	3.42	-14.23	1.23	-25	8
M-w	10	10.50	5.36	1.69	-14.33	-6.67	-21	-4

**Table 1: Descriptive Statistics of Anxiety and Gender Combination**

Figure 2 presents the graph for the anxiety level split by gender combination. It shows that the anxiety level of the pair man-man is the highest followed by woman-woman, then woman-man, and finally man-woman. Interestingly, this rank order of gender combinations in terms of anxiety felt differed markedly from that found in the physical world.



**Figure 2: Anxiety means and 95% confidence intervals for gender combinations.**

The findings from this experiment suggested that avatar gender combination had an influence on the personal space in the CVEs, the ranking of avatar gender combination groups had a striking difference from those observed for personal space invasion in physical environments, and the participants in general did not register high anxiety as might be expected from personal space invasion in physical environments.

#### 4. THE ENVIRONMENT LAYOUT EXPERIMENT

This experiment was conducted to investigate whether the environmental factors that influence personal space in the physical world remain valid in the CVEs. Within the physical world, personal space is influenced by environmental factors such as room size, room ceiling height, and environment location (i.e. indoor or outdoor). Research studies show that the interpersonal distance is increased in small rooms compared with large rooms [Eva96], low ceiling rooms compared with high ceiling rooms [Coh84a], and indoor environments compared with outdoor environments [Coh82b].

In order to investigate how such environmental factors might affect PSI in a virtual world, 8 participants were "invaded" in differently already designed rooms (and also "outdoors") within a CVE and subsequently interviewed about their experiences, see Figure 3 for the virtual rooms.



**Figure 3: The environment for the "Environment Layout" experiment**

Results from this experiment suggested that the reaction to PSI of some participants (3/8) was strong anxiety. Indeed, one participant stopped the experiment and left the environment because of close proximity. Other participants (2/8) reported mild anxiety and the rest of the participants (3/8) reported no anxiety. All of them expressed a preference to keep a distance between their avatars during conversation.

The following are some excerpts from some of the online interviews where "Exp" stands for experimenter and "Part" stands for participant.

**Exp:** what was your feeling when I was close to you?

**Part 5:** well .it feels like in real life. U were a bit to close to me. we do not stand that close in AW [ActiveWorlds] and chat normally. We stand in this kind of distance like now as i real life

(Participant No 7 ran away in the middle of the experiment, even though the conversation had not finished)

**Exp:** why you run away?

**Part 7:** because you were standing too close.

**Part 7:** standing so close makes me uncomfortable

**Exp:** what was your feeling when I was close to you?

**Part 8:** People never stand close like that

Importantly, though, the data suggests that the virtual environment's layout has no effect on the personal space invasion anxiety level, contrary to what would be expected from equivalent research in the physical world in which personal space is influenced by environmental factors such as room size, room ceiling height, indoor or outdoor. The proportion of people expressing anxiety does appear higher in this experiment than the previous experiment. A possible explanation is the difference between the two virtual environments for the experiments. The virtual house of the "Avatar Gender" experiment appeared furnished and the virtual house for this experiment appears completely unfurnished. This is an environmental difference that may influence the perceptions of the users and hence the level of anxiety they felt. Another explanation for the difference of anxiety reported between this and the previous experiment is that the participants of this experiment are anonymous to the experimenter (i.e. the experimenter has not met them in the physical world). This context of anonymity might make the participants behave differently as they are less concerned about the judgement of the experimenter (Fenigstein, Scheier and Buss, 1975). Another possible explanation is that the participants of this experiment were authentic AW users, not students artificially introduced to AW as in the previous experiment.

## 5. DISCUSSION

Caution is needed when considering the above results, for at least the following two reasons. First, participants in the experiments were not able to express their feeling at the time of the invasion, rather they reported it after the experiment, and as such they may have forgotten the exact nature of the feeling when it occurred. Second, ActiveWorlds users can communicate only through text messages – voice messages are not supported. This might also affect the results as the participant might be busy with typing at the keyboard when the confederate's avatar was invading his/her avatar personal space and thus may not notice such an invasion. The switching of attention between the screen and the

keyboard may also reduce the participants' degree of immersion in the environment.

Nevertheless, our results may suggest implications for the design of future collaborative virtual systems interface model. In the physical world people are generally well-rehearsed at avoiding invasions of the space of others, and at taking evasive action if their own space is invaded, for example by turning their orientation away from people who approach too close from the front, or moving physically away. Further, as suggested in section 1 above, many aspects of design in the physical world take account of personal space issues.

If collaborative virtual environments are to be successfully used, it may be necessary to reflect these two approaches to personal space management – reliance on the individual and supportive environmental design – in the virtual environments interface model. Our results suggest that this will not be easy, partly because PSI in a virtual world may not straightforwardly parallel PSI in the physical world, and partly because there appear to be large variations in individual responses to invasions of personal space. Our results also suggest that means of fine grained and easy avatar movement should be incorporated in the virtual environment model, so that (a) they can easily, e.g. via one mouse click, move their avatar away to adopt a new comfortable interpersonal distance and (b) they do not inadvertently invade the personal space of other users (as sometimes happened in the avatar gender experiment).

## 6. CONCLUSION AND FURTHER WORK

The outcomes of two experiments about personal space invasion in CVE have been discussed in this paper. The results of the avatar gender experiment suggested that the participants in general did not register anxiety as in the physical environments. The results of the environment layout experiment suggested that some participants reported high anxiety while the rest either reported low or no anxiety, but there was no evidence of any effect of the environment's layout on levels of anxiety felt.

Further work involves related studies in other virtual worlds, in an attempt to provide a generality test of the present results. In particular, the interaction model of the CVE used for the two experiments does not provide facilities such as voice messaging and facial expressions. Thus, further work in this regard is to investigate the personal space invasion reaction of the users in a CVE that does provide such facilities in its interaction model. Another aspect of our further work is to investigate

whether CVE user experience influences personal space in the CVEs.

The results obtained in this research may to some extent be peculiar to ActiveWorlds, rather than CVEs in general. Therefore, a further research investigating other CVEs would generalise these results. However, the current results remain relevant to any CVE which adopts similar interface policies to ActiveWorlds.

## 7. REFERENCES

[Aie87] Aiello, J. Human Spatial Behaviour Handbook of Environmental Psychology Wiley Inter-science. New York, 1987

[Aie71] Aiello, J. and Jones, E.. Field study of the proxemic behaviour of young school children in three subculture groups, Journal of Personality and Social Psychology, Issue 19, pp 351-356, 1971

[Bail01] Bailenson, J., Blascovich, J., Loomis, J. Equilibrium theory revisited: Mutual gaze and personal space in virtual environments, Presence, Issue. 10, PP 583-597, 2001

[Bax70] Baxter, J. Interpersonal spacing in natural settings, Sociometry, issue 33, pp 444-456, 1970

[Bec98] Becker, B. and Mark, G. Social Conventions in Collaborative Virtual Environments, CVE 98. Manchester, 1998

[Bur98] Burgoon, J., Buller, D., Woodall, G. Nonverbal communication: The unspoken dialogue. New York: Harper and Row, 1998

[Coh82a] Cochran, C. and Urbanczyk, S. The effect of availability of vertical space on personal space Journal of Psychology, Issue 111, pp 137-140. 1982

[Coh84b] Cochran, C., Hale, W., Hissam, C. Personal space requirements in indoor versus outdoor locations, Journal of Psychology, Vol. (117), pages 121-123, 1984

[Dos69] Dosey, M., Meisels, M. Personal space and self-protection Journal of Personality and Social Psychology, Issue 11, pp 93-97, 1969

[Efr87] Efran, M. and Cheyne, J. Affective concomitants of the invasion of shared space: Behavioral, physiological, and verbal indicators, Journal of Personality and Social Psychology, Issue 29, pp 219-226, 1974

[Eli75] Elzinga, R. Nonverbal communication: Body accessibility among the Japanese, Psychologia., Issue 18, pp 205-211, 1975

[Eva96] Evans, G., Lepore, S., Shroeder, A. The role of interior design elements in human response to crowding, Journal of Personality and Social Psychology, Issue 70, pp 41-46, 1996

[Gif96] Gifford, R. Environmental Psychology. Boston MA: Allyn and Bacon. 1996

[Hay83] Hayduk, L. Personal space: Where we now stand Psychological Bulletin, pp 293-335, 1983 Handbook of environmental psychology. John Wiley & Sons, Inc: New York. 2002

[Kri00] Krikorian, D., Lee, L., Chock, T., Harms, C. Isn't that spatial? Distance and communication in a 2 D virtual Environment, Journal of Computer-Mediated Communication [online] 4. 2000

[Hew87] Hewitt J. and Henly R. Sex differences in reaction to spatial invasion. Perceptual and Motor skills, issue 64, pp 809-810, 1987

[Jef98] Jeffrey, P. Personal Space in a Virtual Community, Human Factors in Computing Systems (CHI '98 Summary), Los Angeles. 1998

[Lat95] Latane, B., Liu, J. Nowak, A., Bonevento, M., Zheng, L.. Distance matters: Physical space and social impact, Personality and Social Psychology Bulletin, Issue. 21 (8), pages 795-805, 1995

[Pat87] Patterson, M., and Edinger, J. A functional analysis of space in social interaction ,Siegman A. W. and Feldstein S. (eds), Nonverbal behaviour and communication. Hillsdale, NJ: Erlbaum. PP 523-561, 1987

[Som02] Sommer, R. Personal space in a digital age, In: Bechtel R. B. and Churchman A. 2002

[Som58] Sommer, R. and Ross, H. Social interaction in geriatrics ward, International Journal of Social Psychiatry. Issue 4, 128-133, 1958

[Wil78] Wiles, W. Reassessing personal space in classroom Southern Journal of Educational Research, Issue 12, pp 111-114, 1978



# Using simplified meshes for crude registration of two partially overlapping range images

Mercedes Rocío Gonzales Márquez  
Computer Science Course  
State University of Matogrosso do Sul  
P.O.Box 351, 79804-970 – Dourados, MS, Brazil  
mercedes@uems.br

Wu, Shin - Ting  
School of Electrical and Computer Engineering (FEEC)  
State University of Campinas (Unicamp)  
P.O.Box 6101, 13083-970 – Campinas, SP, Brazil  
ting@dca.fee.unicamp.br

## ABSTRACT

We present an efficient algorithm for crude registration of two partially overlapping range images. The central focus of our work is how to filter out the wrong pairs efficiently in order to reduce the search space. Motivated by the availability of robust and efficient mesh simplification algorithms, such as the QSLIM software package, which preserve the main characteristics of the surface shape, we propose to find a good initial transformation estimate on the basis of the vertices of simplified meshes, instead of the original dense sets of range data. Promising results, that make the iterative closest point matching procedure converges to a correct final alignment, are provided.

**Keywords** Crude registration, Range Images.

## 1 INTRODUCTION

A range image is a two-dimensional grid of depth measurements of a sampling of surface points presented in a scene. As the depth of each imaged point in a given range image is measured from a plane or a single point on the range sensor that captured it, the coordinates of the acquired data are usually represented in terms of the position of this reference. Nevertheless, because of the three-dimensionality of most of objects of interest, this acquisition procedure cannot completely cover an object. Thus, for getting a 3D data model, the capture of several images acquired from different points of view is necessary, as illustrate Figures 1 and 2, and these images may need to be calibrated to a common (global) reference system in a process called *views alignment* or *registration*.



Figure 1: Range images of an angel.



Figure 2: Range images of a dragon.

The registration problem of two partially overlapping range images, *S*-image and *D*-image, may be split into two coupling problems: *correspondence* and *pose* [BMW05a]. A *correspondence* refers to find out the points or features from *S*- and *D*-images that are matchable, and the *pose* problem consists in determining the rigid transformation  $T$  (rotation  $R$  and translation  $t$ ) that may need to be applied on the range data to overlap the matchable points in a common reference system.

Mathematically, for finding the rigid transformation  $T$ , represented by a  $4 \times 4$  matrix, at least three matchable pairs  $(p_i, q_i)$  are required, being  $p_i$  samples of *S*-image and  $q_i$  samples of *D*-image. These pairs must satisfy the equation  $q_i = T p_i$ . If  $N$  pairs are known,  $N > 3$ , the registration problem is, usually, solved by the least squared method  $\min f$  where  $f = \frac{1}{N} \sum_{i=1}^N \|q_i - T p_i\|$ . However, the matchable points are, in most cases, not known. We must estimate them, evaluate the estimated matchable points on the basis of a *fitness error* criterion, and carry the process out iteratively, until a solution with a *fitness error* under the pre-specified tolerance is achieved. Figure 3 shows the registered images of the image pairs presented in Figures 1 and 2.

With respect to the initial estimate, we may classify the existing automatic registration algorithms into two groups: a one-step and a two-step technique. A one-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Copyright UNION Agency – Science Press, Plzen, Czech Republic.

step method aims at obtaining a precise transformation *T en masse*, taking all potential matchable points into consideration. Thus, the problem is dealt with as a 6-dimensional optimization problem. Some researchers apply the genetic algorithm (GA) to find its global minimum, which is theoretically the best solution [BS96, RF02]. To overcome the low convergence of such algorithms, a hybrid genetic algorithm (HGA) that combines the GAs with a numerical method, such as the Quasi-Newton (hill climbing) one is proposed [RP96]. Silva et al. also adopt this approach to get an accurate registration [LSB05]. Despite of several efforts, a one-step procedure is still computationally expansive.

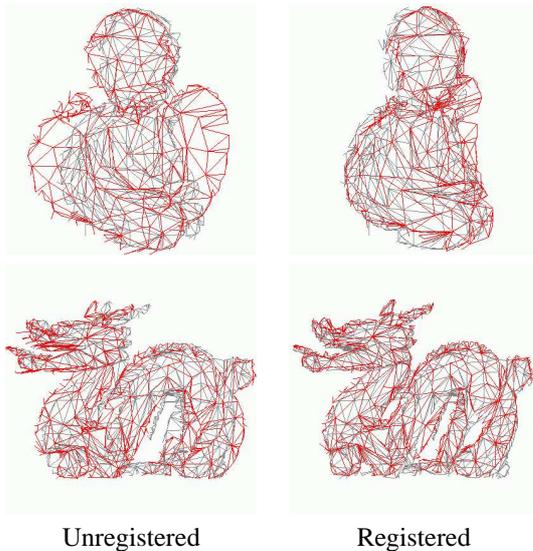


Figure 3: Crude registration of angel and dragon's images.

The basic idea underpinning a two-step method is to handle the problem in two stages: firstly the images are matched from their intrinsic features, then they are finely tuned in order to minimize the point-to-point distance, or the extrinsic, errors. These two steps are, respectively, called the crude registration and the fine registration. From numerical point-of-view, the crude registration, on the basis of the complete collection of samples, provides a good initial estimate  $T_0$  to the fine registration. This, in its turn, attempts to iteratively adjust the transformation matrix, in order to minimize the mean squared errors between the supposed correspondences. The iterative closest point (ICP) [BM92, CM92] and its variants are examples of this category of procedures [RL01].

While the ICP algorithm is recognized as an efficient algorithm for the second step, a robust technique for obtaining  $T_0$  is still pursued by the researchers [WG02, BMW05a]. As detailed in Section 2, the crude registration algorithms basically distinguish in the intrinsic geometric properties that they use for comparison purposes and in the size of the data that they process

for getting plausible local correspondences. If no landmarks are provided, it is still a challenging problem how to *automatically* and *efficiently* extract from the dense data sets the information that is effectively essential for correspondences.

In this paper, we propose to reduce the size of data sets by simplifying the *S*- and *D*-images into *S*- and *D*-meshes with much fewer triangles. Figure 1.(b) illustrates a simplified mesh of the original data set depicted in Figure 1.(a). Though the number of vertices is drastically reduced from 24540 to 200, the vicinity of all the vertices of the simplified mesh keeps the property to be locally alike. Hence, we conjecture that a simplified mesh that preserves the global geometric characteristic of the original data suffices for a coarse registration.

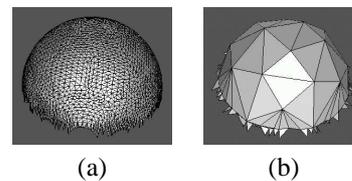


Figure 4: Simplification of a symmetric mesh.

Our proposal is founded on the fact that for crude registration only the geometrically distinguishing features of the input images are of interest. Hence, we may “blur” the local geometrical properties of the input image in such a way that only globally relevant geometric characteristics are kept. From the remaining samples we may get a transformation matrix  $T_0$  that can bring *S*- and *D*-images close enough and that can ensure the convergence of any ICP-like algorithm. We should, therefore, elaborate a mesh simplification algorithm that delivers faithful approximations, namely the *S*- and *D*-meshes, to these dense data.

The *S*- and *D*-meshes may contain distinct number of vertices and possess distinct topology. Comparison strategies, such as the spin-image and the geometric histogram matching, are not applicable in our case. In principle, we may adopt the random sample consensus based data aligned rigidity constrained exhaustive search schema. Nevertheless, after simplification, the size of our input data for registration becomes much smaller and we may exhaustively look for appropriate matchable structures in the *D*-mesh for each structure in the *S*-mesh. This increases the confidence of our search procedure.

Experimentally, we found that the mesh simplification procedure in the QSLIM software package [GP97] satisfies our simplification requirements. A brief description of the algorithm is given in Section 3. The remaining issue is how to construct faithful structure that lead us to a good estimate for  $T_0$ . In Section 4, we provide the geometric metrics that we use to automatically construct the matching structure in the *S*-mesh and the procedure that we adopt to find a matchable one in

the  $D$ -mesh, and consequently, the required alignment transformation. In Section 5 the results that validate our proposal are presented. Finally, in section 6 further work is discussed.

## 2 RELATED WORKS

The algorithms for crude registration work either on the basis of the geometric measurements of the whole sample points with regard to a small set of points, or on the basis of the geometrical characteristics of the vicinity of this small set of points. As far as we know, they work on the dense original range data.

In *Spin-image matching (SIM)* procedure, a small number of oriented points is randomly chosen from  $S$ -image [AM99]. From each of those points, a spin-image is generated, consisting of a 2D-histogram constructed by the radial distance (to the surface normal at the oriented point) and the axial distance (to the tangent plane at the point) of all other points. Then, spin-images associated to distinct points of the  $D$ -image are compared with a few set of the previously built spin-images for constructing matched pairs and estimating transformation candidates. False transformations are pruned by the ICP algorithm. According to the careful comparisons carried out by Planitz et al. [BMW05a], this technique may take a long time and require excessive storage for high resolution surfaces. Moreover, it may fail when the models are axis-symmetric [BMW05b]. Our proposed method not only has less storage complexity but is suitable for axis-symmetric models, as well. We will show that our procedure can successfully register a pair of the axis-symmetric hub's range data presented in Figure 5.

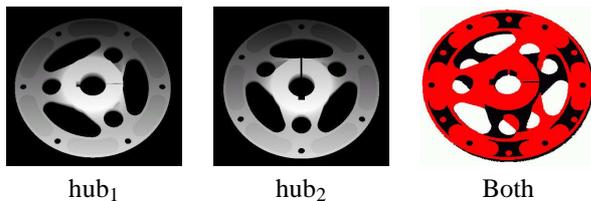


Figure 5: A pair of hub's range images

Instead of histograms constructed by the radial and the axial distances, *Geometric histogram matching (GHM)* method is based on a few set of histograms from the triangular mesh of the  $S$ -image, each of which represents the relative angle between the normal of a given facet  $f_i$  and all other surrounding facets, within a predefined distance from the plane in which  $f_i$  lies to all points of the surrounding facets [AFWR98]. A match for facet  $f_i$  is determined by finding the best match between its histograms and the histograms representing the facets of the triangular mesh of the  $D$ -image. The Random Sample Consensus (RANSAC) algorithm [MR81] is used to compute the best orientation and the best translation for a plausibly correct

final alignment. It was verified that GHM is unsuitable for surfaces with few distinct surface curvature variations [BMW05a]. In this work, we show that our proposed method also works well on models with smooth curvature variations, such as the pairs of images in Figures 6 and 7. In the both cases abrupt curvature transitions are concentrated in a very small region.

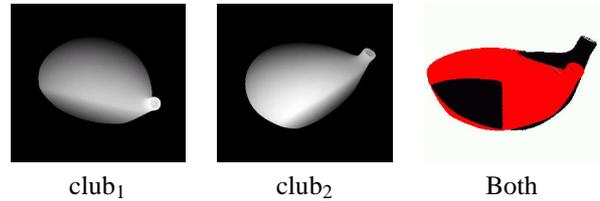


Figure 6: Ranges images of a club.

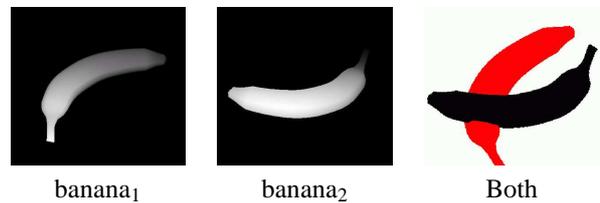


Figure 7: Ranges images of a banana.

*RANSAC-based DARCES (RBD)* is an acronym for Random Sample Consensus based Data Aligned Rigidity Constrained Exhaustive Search [CHC99]. Differently from the SIM and the GHM techniques, it does not require the computation of the geometrical characteristics of the whole sample points. In each iteration of the RBD algorithm, a sample point  $s_1$  is randomly chosen from  $S$ -image, which together with two nearby non-collinear sample points  $s_2$  and  $s_3$  builds a triangular structure  $L$ . Each sample in  $D$ -image is then exhaustively tested aiming to find a triangle that is matchable to  $L$ . For the first point  $s_1$ , any point of  $D$ -image is considered as a potentially matchable point  $m_1$ . The search region for the corresponding point  $m_2$  to  $s_2$  can be limited to the sphere of radius  $d_{12}$  centered at  $m_1$  where  $d_{12}$  is the distance between  $s_1$  and  $s_2$ . And the search region for the corresponding point  $m_3$  to  $s_3$  can be further limited to the intersection of two spheres, one centered at  $m_1$  with radius  $d_{13}$  and the other centered at  $m_2$  with radius  $d_{23}$ . For evaluating the quality of alignment transformation, the authors use a subsample of the original data. Although it works for a wide variety of surfaces, it may take a long time for finding a good initial estimate that converges to a correct alignment. Instead of a planar structure, we propose to use a spatial structure. Because a spatial structure is more distinguishable, fewer matchable structures are selected, which makes feasible to test all of them for getting the best one. Although the modified RBD algorithm implemented by Planitz et al. successfully matches the DINO surface pair presented

in Figure 8 [BMW05a], the outcome of our proposed algorithm is quantitatively and qualitatively better.

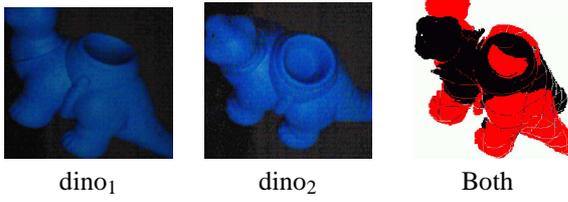


Figure 8: Ranges images of a dinosaur.

In the *Intrinsic curve matching (ICM)* procedure, curves with zero mean curvature are extracted from range images [KPH02]. For each individual range image, all possible pairings between curves are established. In each pairing, the shortest line segment (connector) is determined in *S*-image and its length is used for finding the best correspondence in *D*-image. The quality of the transformation is evaluated by the number of feature alignments. According to Planitz et al. [BMW05a], it works better on surfaces with smooth curvature variations than those with sharp edges. Thus, it fails to perform a good alignment for the pair of images of an object with hard edges, such as the images in Figure 9. Contrary to the ICM algorithm, our proposed model does not suffer from this restriction.

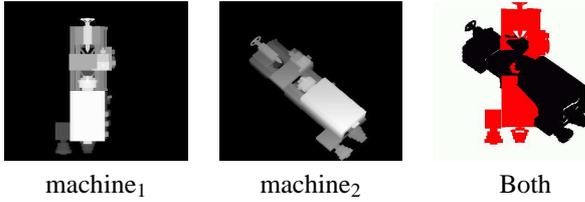


Figure 9: A pair of a machine's images.

### 3 DATA SIMPLIFICATION

To be self-contained, we provide a brief description of the efficient simplification algorithm implemented in the QSLIM package. It uses iterative edge contractions on the basis of quadric errors associated to each vertex  $v$ .

The squared distance of  $v$  to an adjacent face  $k$ , represented by  $n_k \cdot v = d$ , may be expressed as

$$\begin{aligned} D^2(v) &= (n_k^t v + d)^2 = (v^t + d)(n_k^t v + d) \\ &= v^t (n_k n_k^t) v + 2dn_k^t v + d^2 \end{aligned} \quad (1)$$

which may be written in a quadratic form

$$Q(v) = v^t A v + 2b^t v + c = D^2(v)$$

by making  $A = n_k n_k^t$ ,  $b = dn_k$ ,  $c = d^2$ .  $Q$  can be defined as a triple  $Q = (A, b, c)$  where  $A$  is a  $3 \times 3$  matrix,  $b$  is a 3-vector, and  $c$  is a scalar. The addition of quadrics is performed componentwise:  $Q_i(v) + Q_j(v) = (Q_i + Q_j)(v)$  where  $(Q_i + Q_j) = (A_i + A_j, b_i + b_j, c_i + c_j)$ . The

quadric error  $E$  associated to  $v$  is given in terms of the sum of squared distance to all its adjacent faces

$$E(v) = \sum_k D_k^2(v) = \sum_k Q_k(v) = Q(v).$$

Observe that the initial quadric error estimate for each vertex is 0, since it lies in the planes of all its adjacent faces.

Let  $e_{ij}$  be edges of a triangular mesh whose extreme vertices are  $v_i$  and  $v_j$  and  $Q_i$  and  $Q_j$  be their associated quadrics, respectively. In each iteration of the simplification algorithm, the optimal contraction  $\bar{v}$  for each edge is computed (Figure 10), such that the quadric error of its substitute  $\bar{v}$ ,  $Q(\bar{v}) = Q_i(\bar{v}) + Q_j(\bar{v})$  has the minimal quadric error. Since  $Q(\bar{v})$  is quadratic, finding its minimum is a linear problem. The minimum occurs where  $\partial Q / \partial x = \partial Q / \partial y = \partial Q / \partial z = 0$

$$\Delta Q(v) = 2Av + 2b = 0$$

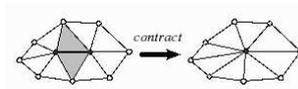


Figure 10: Edge contraction

Assuming that  $A$  is invertible, one gets  $\bar{v} = -A^{-1}b$ . Among all the computed quadric errors, the edge that presents the lowest quadric error is selected to be contracted and the quadric errors of the adjacent vertices are updated. The process is repeated, until the desired level of simplification is reached.

This simplification process satisfies our filtering requirement, which is to “blur” the geometric properties, or to cluster the samples into a vertex, such that the global prominent shapes are preserved, that is, a quadric error of the substitute is less than the specified tolerance.

Figure 3 presents the simplified mesh of a face with several prominent shapes. Observe that, even with a reduced number of vertices, from 3717 to 200, we may still distinguish the nose, eye, chin, cheek, and the hair's comb.

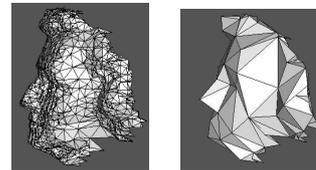


Figure 11: Simplification of a mesh with prominent features.

### 4 CRUDE REGISTRATION

Using a pair of simplified meshes to determine a possible alignment transformation reduces the search space for potential correspondences. Hence, it is expected

that the efficiency of the process is improved. Now, it remains to show how to derive the transformation matrix  $T_0$  from the two simplified meshes. We may adopt either a global optimization method, that takes into consideration the complete collection of vertices of the simplified meshes and involves the solution of a larger system of equations, or a locally aware approach by defining spatial structures with fewer vertices in the  $S$ -mesh and looking exhaustively its pair in the  $D$ -mesh. Aiming at efficiency and robustness, we present in this section a solution for the second approach. The basic idea is to explore the sparsity of the simplified meshes and perform a computationally efficient global search among all of their vertices.

#### 4.1 Defining Matching Structures

Once any isometric transformation in  $\mathfrak{R}^3$  may be characterized by three linearly independent vectors, we propose to use a spatial structure for pairing the vertices of  $S$ - and  $D$ -meshes: a vertex  $v$  of  $S$ -mesh, together with three of 1-neighborhood adjacent vertices (Figure 12(b)).

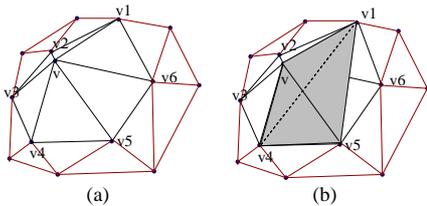


Figure 12: Spatial structure extracted from a 1-neighborhood

Instead of randomly selecting a point in  $S$ -mesh for performing the matching procedure, we use all the vertices whose associated solid angle is above a threshold. In our implementation, it is fixed in  $80^\circ$ . This ensures that no potential matchable pair is missed. For simplicity, we approximate the solid angle of a vertex  $v$  by the median of the angles that all of its incident edges build with its normal vector.

For obtaining the normal vector at  $v$ , we determine the orthogonal regression plane which minimizes the axial distance of the neighboring vertices of  $v$  to it [Smi02]. This regression plane is a good approximation to the tangent plane of the  $S$ -mesh at  $v$ . The vector perpendicular to this plane is, hence, the normal vector of  $S$ -mesh at  $v$ . Let  $v_i$ ,  $i = 1, \dots, m$ , be the  $m$  vertices of the 1-neighborhood of  $v$ . The covariance matrix  $3 \times 3$  in  $v$  is defined as  $\mathcal{C} = \frac{1}{m} \sum_{i=1}^m (v_i - \bar{v})(v_i - \bar{v})^t$ , where  $\bar{v} = \frac{1}{m} \sum_{i=1}^m v_i$  is the mean position vector. The two eigenvectors, corresponding to the largest eigenvalues, define the regression plane and the third eigenvector is, consequently, its normal vector.

It is worth remarking that, differently from the previous crude registration works, our structure encodes two important intrinsic geometric properties of the vertex  $v$ :

area, approximated by the distances between the adjacent vertices, and curvature, approximated by the angle between the normal vector and the adjacent edges. Hence, we believe that our structure is much more discriminant, in the sense that the number of possible pairings is smaller, without degrading the matching quality.

#### 4.2 Pairwise Local Matching

After the simplification procedure, the number of remaining vertices in  $D$ -mesh is small enough that we can exhaustively look for all possible pairings. It is worth remarking that the QSLIM simplification procedure ensures that regions of similar geometric characteristics are similarly contracted, but does not ensure that connectivity of the edges in these regions is the same. To circumvent this problem, we consider for each vertex  $v$  in  $D$ -mesh all the sample points in  $D$ -image that lie in its 4-neighborhood, when we attempt to find the same tetrahedral structure of  $S$ -mesh in the  $D$ -image (Figure 13). Our search procedure follows the same search space reducing principle of the RBD procedure described in section 1.

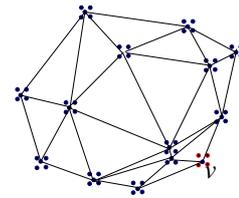


Figure 13: Matchable sample points in  $D$ -image.

Let  $s_i, i = 1..4$  be the vertices of a tetrahedral structure of  $S$ -mesh and let  $d_{ij}$  be the distance between the vertex  $s_i$  and vertex  $s_j$ . The corresponding sample point  $m_1$  of  $s_1$  must be in the 4-neighborhood of the  $s'_1$ 's correspondence in  $D$ -image. The correspondences  $m_2$  and  $m_3$  to  $s_2$  and  $s_3$ , respectively, are obtained in the same way as in the RBD algorithm. And the  $m_4$ , that corresponds to  $s_4$ , must be the intersection of three spheres: the first centered in  $m_1$  and with radius  $d_{14}$ , the second with center in  $m_2$  and radius  $d_{24}$  and the later, centered in  $m_3$  and with radius  $d_{34}$ . Figure 14 illustrates the search spaces for the four correspondences in  $D$ -image.

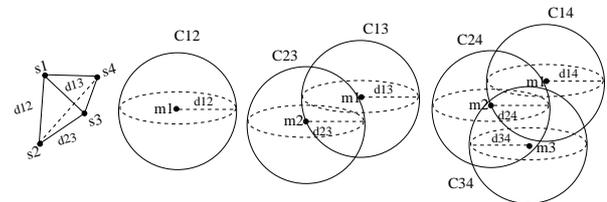


Figure 14: Pairwise matching.

If the correspondences of  $s_i$  are found, we further compare their solid angles. If they are approximately the same, then we consider that  $(s_1, s_2, s_3, s_4)$  and  $(m_1, m_2, m_3, m_4)$  are paired off.

Images	Planar Struct.	Spatial Struct.	% Reduction
Figure 1	11,182,373	1,927,148	82,77%
Figure 2	51,661,440	8,739,339	83,08%
Figure 15	35,402,841	3,242,086	90,84%
Figure 5	125,661,822	3,303,893	97,37%
Figure 6	15,237,324	289,286	98,10%
Figure 7	12,190,662	1,157,796	90,50%
Figure 8	9,253,616	1,391,281	84,97%
Figure 9	24,001,344	909,067	96,21%

Table 1: Planar  $\times$  spatial efficiency comparison

Images	Local Pairings	% Filtering	Iterations ICP
Figure 1	1,927,148	92.38%	6
Figure 2	8,739,339	84.42%	7
Figure 15	3,242,086	73.29%	8
Figure 5	3,303,893	83.32%	7
Figure 6	289,286	89.41%	5
Figure 7	1,157,796	92.72%	4
Figure 8	1,391,281	93.08%	5
Figure 9	909,067	94.00%	8

Table 2: Similarity and visibility efficiency.

To support our remark in Section 4.1, we compared experimentally the filtering efficiency of a planar structure used in the RBD procedure [CHC99] and our proposed spatial structure. Table 1 summarizes in the column 2 and 3, respectively, the number of matchable samples for planar and spatial structures. Observe that the spatial one presents higher filtering efficiency in all of cases.

### 4.3 Additional Filterings

Before estimating the transformation  $T$  that aligns all the range data, we may still filter out some wrong local correspondences that we obtained from the structure matching. There are two trivial tests: neighborhood similarity and visibility tests. In the former, we determine the distance of the aligned points in the vicinity of a matched pair. If it is greater than a pre-specified tolerance, we may discard the correspondence. The latter is based on the fact that if one triangle is on overlapping region, it is visible from the  $S$ -viewpoint, then after alignment it must also be visible from the  $D$ -viewpoint. We apply the visibility test to all 1-neighborhood faces of the locally pairing vertices. If less than 50% of faces pass the test, the correspondence is discarded. Table 2 shows the percentage of the number of matchable points that is reduced after the trivial filtering for the pairs of images presented in this paper. The column Local Pairings provides the number of matchable tetrahedral structures before filtering and the column % Filtering shows the reduction ratio, after filtering, which is in average 87.82%.

## 5 EXPERIMENTAL RESULTS

In this section we present some results that corroborate our conjecture: faithful approximations to dense range images may be used for finding a good correspondence estimate for two partially overlapping range data sets. We apply our algorithm on the data that are similar to the ones employed by Planitz et al. for analyzing and comparing the most known surface correspondence algorithms, namely the spin-image (SIM), the geometric histogram (GHM), the RANSAC-based DARCES (RBD), and the intrinsic curve (ICM) matching procedures. These data vary in acquisition, size, level of detail, and geometric characteristics [BMW05a].

The ANGEL surface pair (Figure 1) and the DINO surface pair (Figure 8) are the same that Planitz et al. used in their work. We also include the FROG surface pair from the Ohio State University repository [OSU] that are  $60^\circ$  apart from each other (Figure 15) and the DRAGON surface pair (Figure 2), scanned with a laser triangulation Cyberware 3030 MS scanner [Sta] and of high level of details.



Figure 15: Range images of a frog.

The rest of surface pairs have been obtained from the Stuttgart repository [Stt]. The CLUB surface pair (Figure 6) was also captured by a Cyberware scanner. It has abrupt curvature variations concentrated in a very small region. The BANANA surface pair (Figure 7) and the MACHINE image pair (Figure 9) are synthetic images from the 3D Cafe website [3dc]. The BANANA images also contain very smooth curvature variations, while the MACHINE surface has sharp edges and planar facets.

Firstly, we apply the QSLIM simplification algorithm in the original data samples, whose number is given in the column Samples of Table 3. The number of vertices of the outcomes are listed in the column Vertices. From these outcomes, our proposed algorithm only selects a few number of tetrahedral structures, as shows the column Structures. The reduction ratio is more than 99.5%. Despite this drastic reduction, our algorithm is able to deliver alignment matrices for the seven surface pairs that make the ICP algorithm converges. Figure 16(a) presents the simplified meshes output by the QSLIM algorithm and Figure 16(b), the visual results of our proposed procedure. These results show that satisfactory crude registration between all the surface pairs is achieved.

According to Planitz et al., we may evaluate quantitatively the performance of our proposal by verifying

Images	Samples	Vertices	Structures
angel <sub>1</sub>	14812	416	54
angel <sub>2</sub>	10632	402	83
bluedino <sub>1</sub>	20494	422	33
bluedino <sub>2</sub>	16783	435	24
frog <sub>1</sub>	9640	297	90
frog <sub>2</sub>	8618	374	56
hub <sub>1</sub>	72099	1039	58
hub <sub>2</sub>	71772	1039	60
banana <sub>1</sub>	21312	542	26
banana <sub>2</sub>	22071	521	24
club <sub>1</sub>	31954	516	22
club <sub>2</sub>	32746	514	25
machine <sub>1</sub>	28086	437	110
machine <sub>2</sub>	33884	552	86
dragon <sub>1</sub>	41841	685	44
dragon <sub>2</sub>	34836	821	80

Table 3: Reduction of samples.

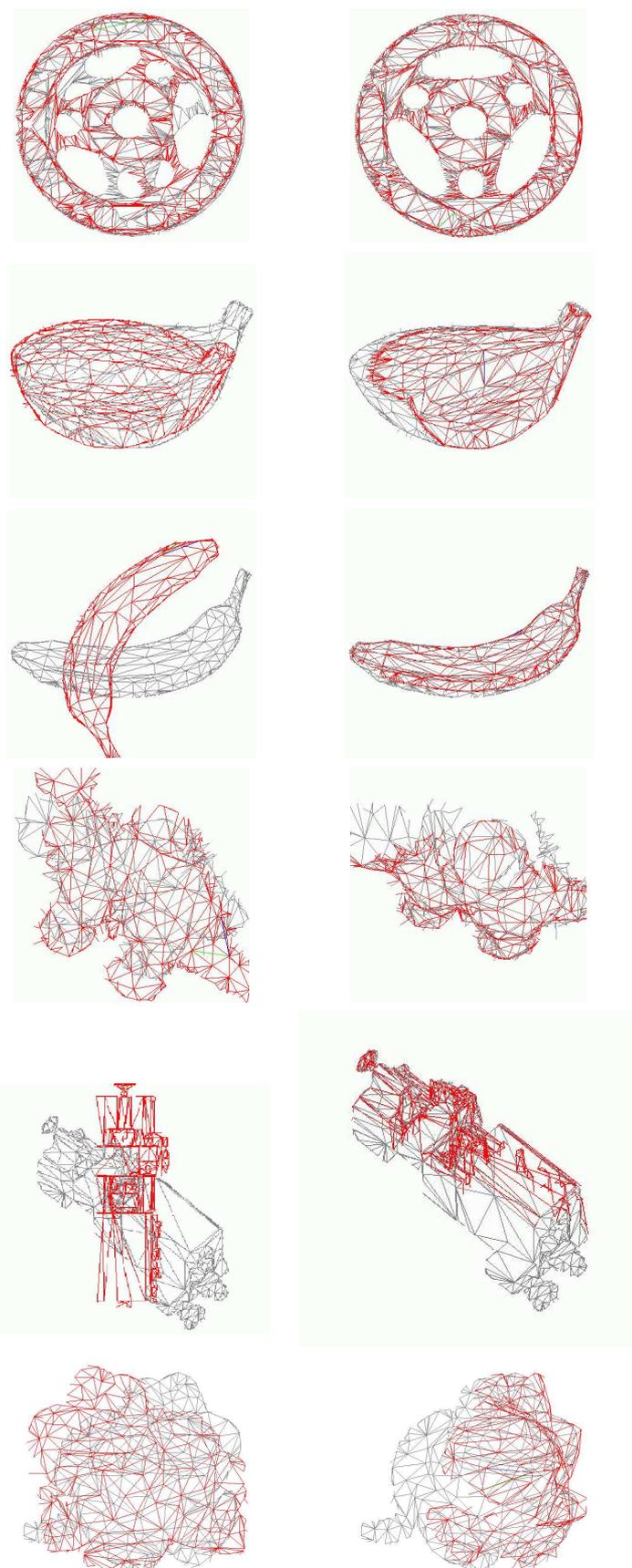
whether its outcomes lead to correct convergence of the ICP procedure. We used the Scanalyze system, developed in the Stanford University [Sca], for evaluating our results. Visually, the outcomes from the Scanalyze is almost indistinguishable from the images shown in Figure 16(b). The column Iterations ICP of Table 2 provides the number of iterations that the ICP algorithm needed to converge to the optimal solution.

## 6 CONCLUDING REMARKS

In this paper we present a technique for an automatic registration of two partially overlapping range images on the basis of their simplified triangular meshes. The most important contribution of our work for crude registrations is to use a faithful approximation of the dense range data. Among the advantages of this approach, we may list: (1) drastic reduction of the input data, without missing the relevant features, (2) easiness in constructing discriminant structures that integrate two intrinsic properties: distance and curvature, and (3) simplicity in defining additional trivial discard rules for pruning false transformations. As further work, we would like to evaluate the sensitive of our method with regard to the threshold of the solid angle and to reconstruct a complete 3D model from the registered range data.

## 7 ACKNOWLEDGMENTS

We would like to acknowledge FAPESP for financial support under the grant numbers 00/10913-3 and 2003/13090-6.



(a) Simplified mesh (b) Coarse registration  
Figure 16: Crude registration results

## REFERENCES

- [3dc] Range images. <http://www.3dcafe.com/>.
- [AFWR98] A.P. Ashbrook, R.B. Fisher, N. Werghi, and C. Robertson. Aligning arbitrary surfaces using pairwise geometric histograms. In *Proceedings of Noblesse Workshop on Non-linear Model Based Image Analysis*, pages 103–108, 1998.
- [AM99] A.Johnson and M.Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:433–448, 1999.
- [BM92] P.J. Besl and N.D. McKay. A method for registration of 3-D shapes. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.
- [BMW05a] B.M.Planitz, A.J. Maeder, and J.A. Williams. The correspondence framework for 3d surface matching algorithms. *Computer Vision and Image Understanding*, 97:347–383, 2005.
- [BMW05b] B.M.Planitz, A.J. Maeder, and J.A. Williams. Using the correspondence framework to select surface matching algorithms. In *Proceedings of the Australian Pattern Recognition Society (APRS) Workshop on Digital Image Computing*, pages 85–90, 2005.
- [BS96] K. Brunnstrom and A.J. Stoddart. Genetic algorithms for free-form surface matching. In *Proceedings of 13th International Conference on Pattern Recognition*, volume 4, pages 689–693, 1996.
- [CHC99] Chu-Song Chen, Yi-Ping Hung, and Jen-Bo Cheng. RANSAC-based DARCES: A new approach to fast automatic registration of partially overlapping range images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1229–1234, 1999.
- [CM92] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. *Image and Vision Computing*, pages 145–155, 1992.
- [GP97] M. Garland and P.Heckbert. Simplification using quadric error metrics. In *Computer Graphics*, volume 31, pages 209–216, New York, USA, 1997.
- [KPH02] Pavel Krsek, Tomáš Pajdla, and Václav Hlaváč. Differential invariants as the base of triangulated surface registration. *Computer Vision Image Understanding*, 87:27–38, 2002.
- [LSB05] O.R.P. Bellon L. Silva and K.L. Boyer. Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms. *IEEE Transactions on Pattern and Machine Intelligence*, 27:762–776, 2005.
- [MR81] M.A.Fischler and R.C.Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. Association for Computing Machinery*, 24(6):381–395, 1981.
- [OSU] Range images. <http://sampl.eng.ohio-state.edu/sampl/data/3DDB/RID>.
- [RF02] C. Robertson and R.B. Fisher. Parallel evolutionary registration of range data. *Computer Vision and Image Understanding*, 87:39–50, 2002.
- [RL01] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Third International Conf. on 3D Digital Imag. and Modeling*, pages 145–152, 2001.
- [RP96] J-M. Renders and Stéphane P.Flasse. Hybrid methods using genetic algorithms for global optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, 26:243–258, 1996.
- [Sca] Scanalyze, a system for aligning and merging range data. <http://graphics.stanford.edu/software/scanalyze>.
- [Smi02] L.I. Smith. A tutorial on principal components analysis. [http://www.cs.otago.ac.nz/COSC453/student-tutorials/principal\\_components.pdf](http://www.cs.otago.ac.nz/COSC453/student-tutorials/principal_components.pdf), Feb. 2002.
- [Sta] Range images. <http://graphics.stanford.edu/data/3Dscanrep>.
- [Stt] Range images. <http://range.informatik.uni-stuttgart.de/htdocs/html>.
- [WG02] Joris Vanden Wyngaerd and Luc Van Gool. Automatic crude patch registration: Toward automatic 3d model building. *Computer Vision and Image Understanding*, 87(1–3):8–26, July 2002.

# An Image Matching Using Critical-Point Filters and Level Set Analysis

Kittipong Thanasoontornlerk  
Department of Computer Engineering  
Faculty of Engineering  
Chulalongkorn University  
4870221721@student.chula.ac.th

Pizzanu Kanongchaiyos  
Department of Computer Engineering  
Faculty of Engineering  
Chulalongkorn University  
pizzanu@cp.eng.chula.ac.th

## ABSTRACT

Critical-Point Filters (CPFs) are nonlinear filters which preserve intensity and location of each critical point in the image and reduce the resolution without any prior knowledge. Although CPFs can avoid blurred intensity and ambiguous location problem of previous linear filters, its computational cost is still expensive due to its complexity. We propose an enhancement of the CPFs algorithm for image matching using level set analysis. An image is analyzed and transformed to hierarchical level sets of pixel having same intensity. Connectivity of the level sets represents the image contrast invariant features. Between the corresponding level sets of the input images, two pixels are mapped based on their energy and bijectivity conditions. Finally, less computational time with precise mapping is shown in the experimental result.

## Keywords

Image morphing, Feature measurement, Level set, Critical-point filter

## 1. INTRODUCTION

Image matching is an important problem in many researches such as computer graphics, computer animation and computer vision [BJNS03, Jcu02a, Jcu03a, Jdu02c]. However, there isn't efficient algorithm for matching images because algorithms for image matching usually trade off between complex computation and accuracy. Image Filter is a favored method for matching because the principle of image filter is the enhancement and attenuation of image attributes such as edges or the intensity of the image. Afterwards, Images are matched using image attributes. Filters are classified into two groups: linear filters and non-linear filters. The Linear filters are not useful because of its blurred intensity and ambiguous location. For example, the stereo photogrammetry methods use the edge detection which is high-pass filter. This method suffers from noise because high-pass filters can also detect noise. [Chr96a, YT98].

Multiresolutional Critical Point Filters (CPFs) are non-linear filters which effectively and accurately match two images. This filter extracts a maximum, a minimum, and two types of saddle point of pixel intensity for every small region which the results are four multiresolution hierarchies. The matching of the images is computed in each hierarchy, each resolution, from coarse level to the fine level.

CPFs are the same concept as an image pyramid [Fra01a, RR02a]. Image pyramid is multiple resolution of the original image which is reduced by a factor of 2. Figure 1 and Figure 2 show examples of image pyramid.

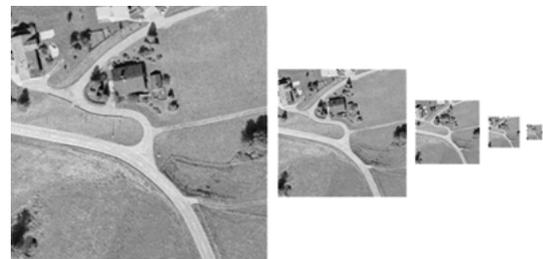


Figure 1. Example of an image pyramid [Chr96a]

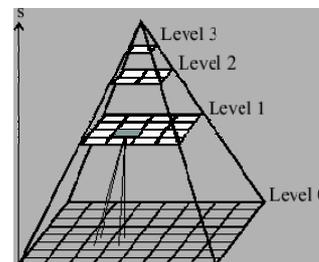
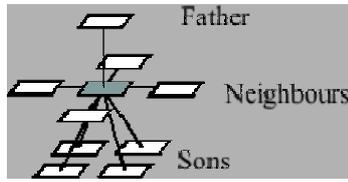


Figure 2. Example hierarchy of image pyramid [Fra01a]

The structure of an image pyramid is determined by 2 directions: The horizontal relations between neighborhoods in the same level and, the vertical

relations; parent-child relations, between adjacent levels. Image pyramid combines the advantages of both high and low resolutions of image following table 1.



**Figure 3. Example structure of image pyramid [Fra01a]**

	High resolution	Low resolution
Data amount	huge	small
detail	rich and many	very few
overview	bad	Low
precision	high	Low

**Table 1. Qualities of images at different resolutions [Fra01a]**

However, matching two images by using CPFs requires complicated computation and redundancy of image data because of hierarchy of multiresolution.

In this research, level set analysis [PAD02a] is used for creating regions in each hierarchy. The matching of two images uses iteratively searching corresponding region. The image level set is a hierarchy of the gray intensity level image. An image will be transformed to image level sets. Connected components in each level are found. The components which have the size of connected component less than a pre-defined value will be removed.

Section 2 introduces the definition of CPFs, and Section 3 introduces the definition of image level set. Correspondence level set presents in Section 4. The results of experiment are shown in Section 6. The last section is the conclusion and future work.

## 2. THE MULTIREOLUTIONAL CRITICAL POINT FILTER

Multiresolutional Critical Point Filters (CPF) are introduced by Shinagawa and Kunii [YT98]. CPFs are non-linear filters that extract the critical point of the image in each resolution and do not destroy the essential structures of the critical points in image. The height and the width of the image are equal, power of two. The matching method uses an energy equation; based on pixel intensity and location, to search point correspondence between two images which have minimized energy.

## 2.1 Critical Point Filters

CPFs create four multi-resolution hierarchies that represent minimum, two saddles and maximum point of image; following equation 1 to equation 4. In other words, CPFs extract a maximum, a minimum, and two types of saddle points of intensity pixel for every  $2 \times 2$  (horizontal  $\times$  vertical) pixels. Then, an image of low level of resolution is generated from an image of high level of resolution for each type of a critical point. Let width and height of image be  $N$  and  $M$ , respectively. Thus,  $N = M = 2^n$ . A pixel of image at position  $(i, j)$  is defined by  $p_{(i,j)}^{(m,l)}$ , where  $m$  represents the level of the hierarchy.

$$p_{(i,j)}^{(m,0)} = \min(\min(p_{(2i,2j)}^{(m+1,1)}, p_{(2i,2j+1)}^{(m+1,1)}), \min(p_{(2i+1,2j)}^{(m+1,1)}, p_{(2i+1,2j+1)}^{(m+1,1)})) \quad (1)$$

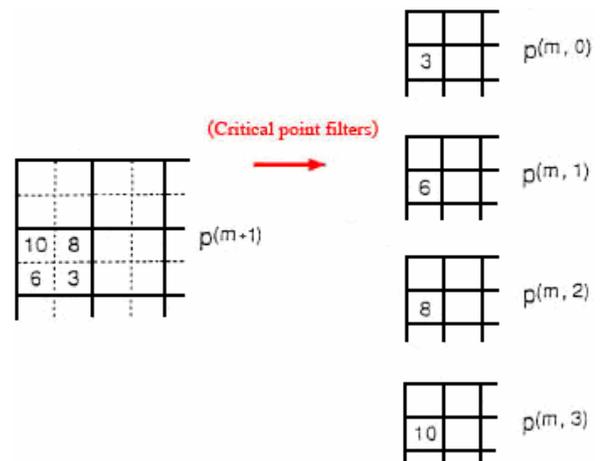
$$p_{(i,j)}^{(m,1)} = \max(\min(p_{(2i,2j)}^{(m+1,1)}, p_{(2i,2j+1)}^{(m+1,1)}), \min(p_{(2i+1,2j)}^{(m+1,1)}, p_{(2i+1,2j+1)}^{(m+1,1)})) \quad (2)$$

$$p_{(i,j)}^{(m,2)} = \min(\max(p_{(2i,2j)}^{(m+1,1)}, p_{(2i,2j+1)}^{(m+1,1)}), \max(p_{(2i+1,2j)}^{(m+1,1)}, p_{(2i+1,2j+1)}^{(m+1,1)})) \quad (3)$$

$$p_{(i,j)}^{(m,3)} = \max(\max(p_{(2i,2j)}^{(m+1,1)}, p_{(2i,2j+1)}^{(m+1,1)}), \max(p_{(2i+1,2j)}^{(m+1,1)}, p_{(2i+1,2j+1)}^{(m+1,1)})) \quad (4)$$

where

$$p_{(i,j)}^{(n,0)} = p_{(i,j)}^{(n,1)} = p_{(i,j)}^{(n,2)} = p_{(i,j)}^{(n,3)} = p_{(i,j)}$$



**Figure 4. Procedure of CPFs**

The four images are called as subimage having size  $\frac{1}{4}$  of the higher level. Figure 4 shows the example of CPFs, which a value in the picture is an intensity value of image.

## 2.2 Hierarchies

Four multiresolution hierarchies at depth  $d$  of each input image are computed, where the size of each image at  $m^{\text{th}}$  level is  $2^m \times 2^m$  ( $0 \leq m \leq n$ ) pixels. Figure 5 shows an example of three hierarchies which original image is  $8 \times 8$  pixels.

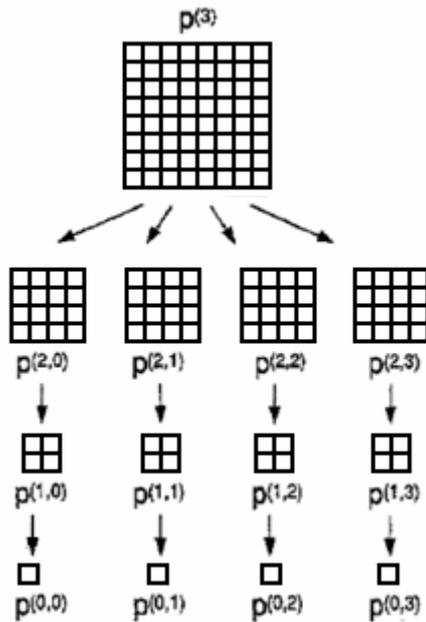


Figure 5. Multiresolutional hierarchy

Figure 6 shows the result of CPFs, Original image is eight-bit gray-scale image having size  $256 \times 256$  pixels. Subimages are size  $64 \times 64$  pixels at 6 levels of hierarchies. Both original image and subimages shows same resolution. In each subimage, checkerboard pattern (or aliasing) happens around the borders of object [RR02a].

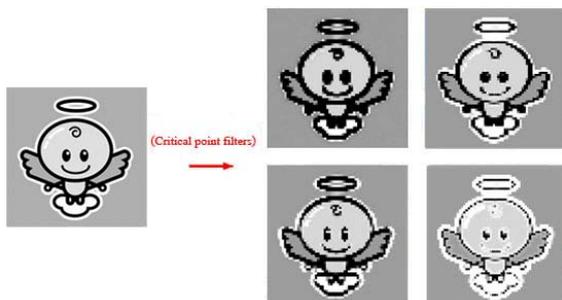


Figure 6. The Original image and subimages at level 6 of the hierarchy (original image is size  $256 \times 256$  pixels and subimage is  $64 \times 64$  pixels)

## 2.3 Image matching

In matching two images, the first image is called the *source image* and the second is called the *destination image*. Let  $p_{(i,j)}^n$  be the pixel of the source image at the position  $(i, j)$ , and  $q_{(k,l)}^n$  be the pixel of the destination image at the position  $(k, l)$ .

Parent-child relationships are defined by:  $p_{(i',j')}^{(m-1,s)}$  at level  $m-1$  is called *parent* of  $p_{(i,j)}^{(m,s)}$  at level  $m$  where  $(i', j') = (\lfloor \frac{i}{2} \rfloor, \lfloor \frac{j}{2} \rfloor)$ . Conversely  $p_{(i,j)}^{(m,s)}$  at level  $m$  is called *child* of  $p_{(i',j')}^{(m-1,s)}$  at level  $m-1$ . The function  $parent(i, j)$  is defined as

$$parent(i, j) = (\lfloor \frac{i}{2} \rfloor, \lfloor \frac{j}{2} \rfloor)$$

Point  $p$  in source image will be mapped to point  $q$  in destination image at level  $m$ . Point  $a, b, c$  and  $d$  are 4 nearest diagonal neighbors of point  $p$  and they are children of  $A, B, C$  and  $D$  at level  $m-1$ . Point  $A, B, C$  and  $D$  of the source image are mapped to  $A', B', C'$  and  $D'$  of the destination image at level  $m-1$ . The pixel  $p$  should be mapped to the pixel  $q$  inside of inherited quadrilateral  $A'B'C'D'$ .

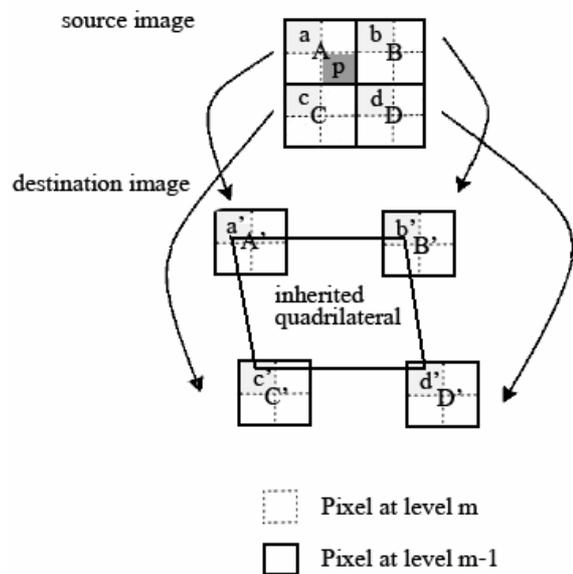


Figure 7. The inherited quadrilateral of pixel  $p$

Finding corresponding point between two images use an energy function determined by the difference in the intensity of the pixel of the source image and its corresponding. Points are mapped recursively by computing and comparing the energy of candidate points located in the inherited quadrilateral

## 2.4 Energy of mapping

Let  $p_{(i,j)}$  be the point to map and  $q_{(k,l)}$  be the point to test within the quadrilateral inheritance. Between two points is computed using energy, find minimum energy, for each subimage and at each level of the hierarchy

### 2.4.1 Cost related to the pixel intensity

The energy determines difference of intensity in source image pixel and its corresponding pixel of the destination image.

$$C_I = \left| V(p_{(i,j)}) - V(q_{(k,l)}) \right|^2$$

where  $V(p_{(i,j)})$  and  $V(q_{(k,l)})$  are the intensity values of the pixel  $p_{(i,j)}$  and  $q_{(k,l)}$ , respectively.

### 2.4.2 Cost related to the locations of the pixel for smooth mapping

This energy is used to prevent a pixel from being mapped too far away and to ensure the smoothness of mapping.  $D$  is determined by the value difference locations.

$$D = \eta E_0 + E_1$$

where  $\eta \geq 0$  is a real number and

$$E_0 = \left\| (i, j) - (k, l) \right\|^2$$

$$E_1 = \sum_{i=i-1}^i \sum_{j=j-1}^j \left\| ((k, l) - (i, j)) - ((k', l') - (i', j')) \right\|^2 / 4$$

where  $\left\| (x, y) = \sqrt{x^2 + y^2} \right\|$ .  $(i', j')$  is defined to be zero for  $i' < 0$  or  $j' < 0$ .

$E_0$  prevent a pixel being mapped to a pixel too far away.  $E_1$  ensure the smoothness of the mapping.

### 2.4.3 Total energy of the mapping

The total energy of the mapping is defined as  $\lambda C_f^{(m,s)} + \gamma D_f^{(m,s)}$ . The matching finds corresponding points which have minimum energy.

$$\min \lambda C_f^{(m,s)} + \gamma D_f^{(m,s)}$$

Where  $\lambda$  and  $\gamma$  are constants (0.1 and 0.01 respectively [Jcu02a])

## 3. IMAGE LEVEL SET

Let  $\Psi$  be the set of all intensity values in gray-scale image. In 8-bits gray-scale image will be  $\Psi = \{0, 1, 2, \dots, 255\}$ . Let  $Z_m = \{1, 2, \dots, m\}$  by  $M$  be height of the image, and  $Z_n = \{1, 2, \dots, n\}$  by  $N$  be width of the image. Let  $\lambda$  be real function represents the spatial information of the image;  $\lambda : Z_m \times Z_n \rightarrow \Psi$ . If  $z$  be a pixel in the image,  $z \in Z_m \times Z_n$ , and the corresponding intensity value  $\zeta \in \Psi$ , then we can define  $\lambda(z) = \zeta$ .

### 3.1 Level set

Level sets [PAD02a] and its properties are defined as following;

**Definition:** For  $\zeta \in \Psi$ , consider the following sets

$\omega$  is a set of intensity which has value more than or equal to  $\zeta$ . Define by

$$\omega_{\zeta \in \Psi} = \{x \in \Psi : x \geq \zeta\}$$

$\theta$  is a set of pixel which has intensity value equal to  $y$ . Define by

$$\theta_{\zeta \in \Psi} = \{z \in Z_m \times Z_n : \exists y \in \omega_{\zeta \in \Psi} \ni$$

$$\lambda(z) = y\}$$

Consider the function  $L_\zeta : Z_m \times Z_n \rightarrow \{0, 1\} \ni$

$$L_\zeta(z) = \begin{cases} 1 & \text{if } z \in \theta_{\zeta \in \Psi} \\ 0 & \text{otherwise} \end{cases}$$

The matrix representation of  $L_\zeta$  is binary image which is called a level set and  $\zeta$  is corresponding level (see Fig.8).

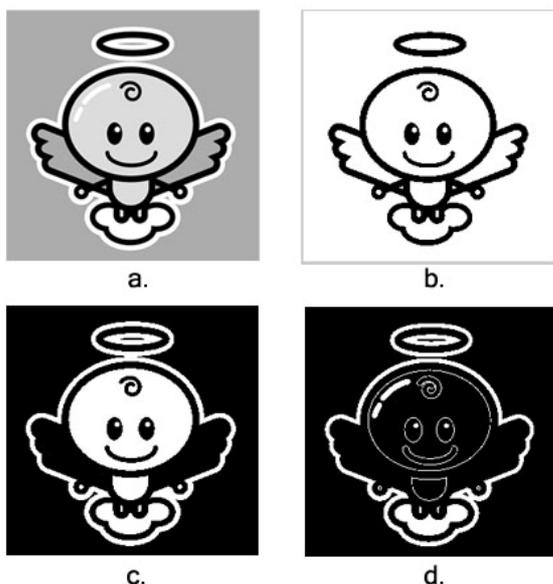
Basic properties of the level sets are also defined as the following :

**Theorem 1.** The total pixel-wise sum of all the level sets is equal to the image.

**Lemma 1:**  $\forall z \in Z_m \times Z_n, \forall \alpha, \beta \in \Psi : \alpha > \beta$  then  $L_\alpha(z) = 1 \Rightarrow L_\beta(z) = 1$  but not conversely.

**Lemma 2:**  $\forall z \in Z_m \times Z_n, \forall \alpha, \beta \in \Psi : \alpha > \beta$  then  $L_\beta(z) = 0 \Rightarrow L_\alpha(z) = 0$  but not conversely

**Lemma 3:**  $\forall z \in Z_m \times Z_n, \lambda(z) = \alpha \in \Psi$ , then  $L_\alpha(z) = 1$  and  $L_{(\alpha+1)}(z) = 0$



**Figure 8. Level sets of an image (a) Original image (b) Level set at intensity 64 (c) Level set at intensity 192 and (d) Level set at intensity 224**

A Level set of 8-bits image can be reachable to 255 levels. In Figure 8, three examples of level sets of image at 64, 192 and 224 are shown respectively. Each figure is a binary image; white represents pixel valued one, and black represent pixel valued zero.

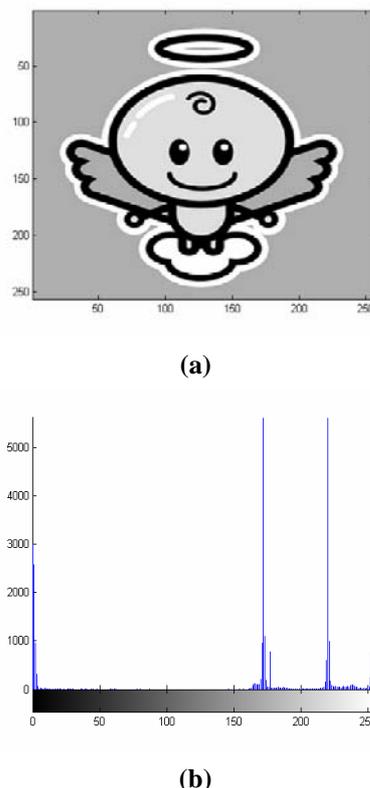
In our method, the corresponding level sets are used for matching instead of corresponding pixel having minimum value of intensity. The period of intensity is considered from histogram of image. The detail is described in Section 4.

#### 4. CORRESPONDANCE LEVEL SET

This section explains how to use level set for reduce computational time of the original CPFs. The level set is initially bounded by a range of intensity whereas its threshold is defined from the histogram of the input image. Figure 9 shows an example image and its histogram which we can classify level of the image into 7 levels concerning about its histogram.

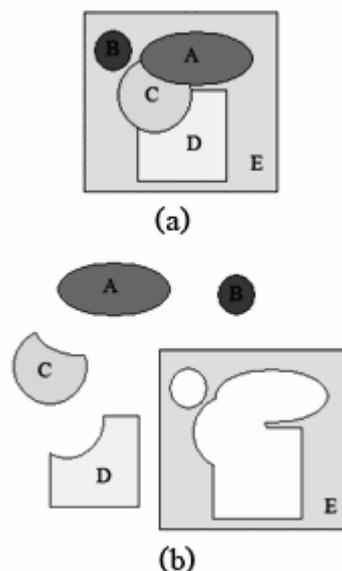
The computation of four multiresolution hierarchies of image following equation 1 to equation 4 is then composed. For each resolution, level set and connecting components of an input image are extracted at the same time.

Matching between corresponding components is then computed by comparing between level sets at each hierarchy of the two images from minimum level to the maximum level. The corresponding level set is then mapped. The mapping algorithm is as follows:



**Figure 9. Computing histogram of an image (a) Original image (b) Histogram of original image**

1. Filter the input images with critical point filters following equation 1 to equation 4.



**Figure 10. Example of extracting image components**

2. Find connecting components in each level set of image for every level of hierarchies. Figure 10 shows an example of input image having five level sets, A B C D and E, which are divided to five connecting components.

3. Map between connecting components of source image and destination image in corresponding level set and level of hierarchies having shortest average distance between corresponding pixels in connect components according to parent – child relationship starting from coarse level to the fine level.

5. Map between pixels in corresponding connecting components by using energy of intensity and energy of location. Figure 11 displays an example of mapping at a fine level having three level sets.

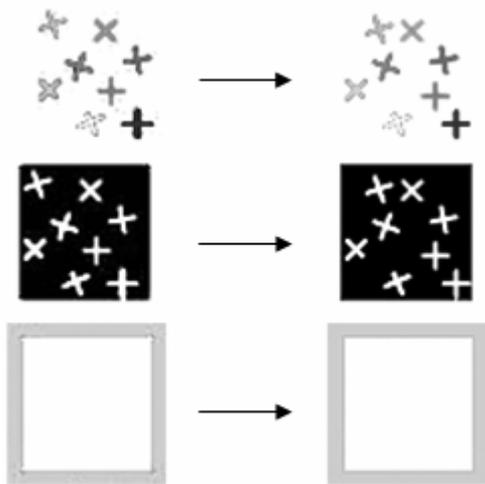


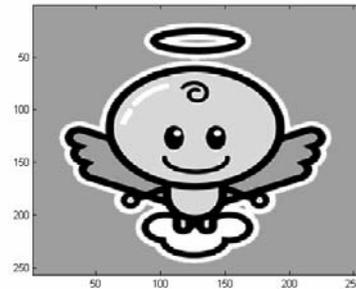
Figure 11. Example of mapping in fine level

## 5. TIME COMPLEXITY

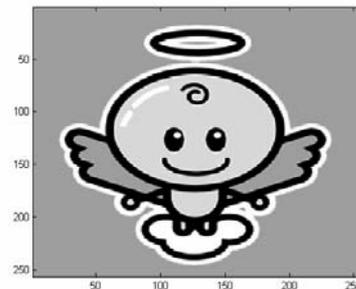
The time complexity of filtering of the image size ( $n*n$ ) is  $\theta(n^2)$ , while mapping between input images can be reduced to mapping between two graphs having an edge between each connecting component in each level set and hierarchical level represented as a node. Although the subgraph isomorphism is categorized in the complexity class *NP-complete*, the number of nodes here depends on the similarity of the input image and the size of level set which is adjustable. Fastest known subgraph isomorphism can test most graphs of less than 100 vertices in well under a second [BD81]. The worst case happens when the size of the graph grows up to  $O(n^2)$  which equals to the original CPFs.

## 6. EXPERIMENTAL RESULTS

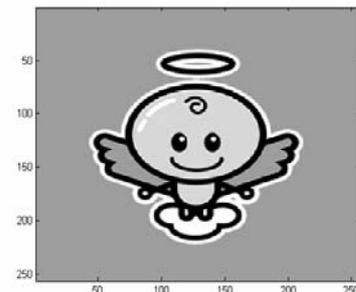
The experiment of image matching based on the proposed algorithm is conducted by MATLAB 7.0. In the experiment the destination images are applied by affine transformations. Figure 12 shows examples of source image and destination image in our experiment. Figure 13 shows a result from our method. Blue pixels in source image are mapped to blue pixels in destination image properly. Figure 14 and 15 graph present correspondence of point between source image and destination image.



(a)

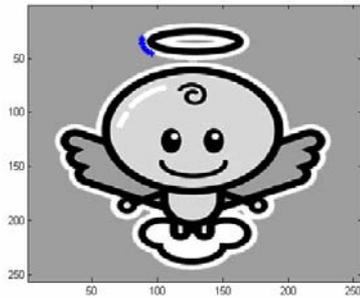


(b)

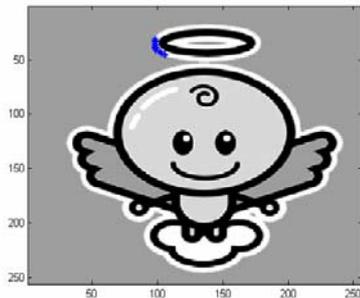


(c)

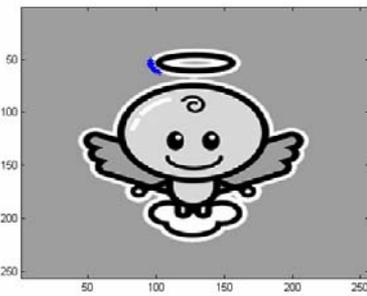
Figure 12. Input images a) source image (b) destination image shifted by 10 pixels in x axis (c) destination image reduce size 20 % of source image



(a)



(b)



(c)

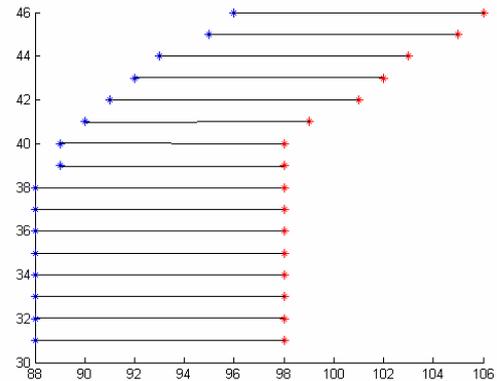
**Figure 13. Result of mapping (a) source image (b) destination image shifted by 10 pixels in x axis and (c) destination image reduce size 20 % of source image**

## 7. CONCLUSION AND FUTURE WORK

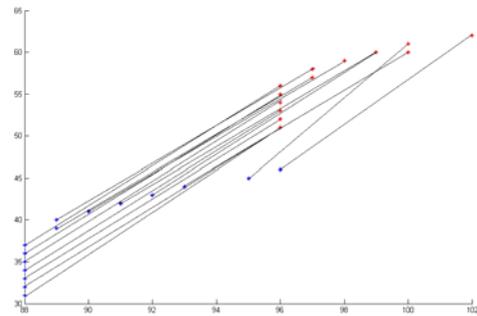
We propose an enhancement of the critical point filters algorithm using level set analysis. Our method can reduce redundancy of an image data in matching image, find corresponding pixels by original CPFs comparisons. The experiment, our method has high accuracy in matching image. This method can apply affine transformation image. Our method is work properly with clearly image, source image and destination image have the same detail in the image, and width and height of image must have equal size and power of two.

More efficient algorithms can be developed based on both faster subgraph isomorphism algorithm and faster level set method. Both matching for animating image and more complex image seem different properties or resolution is also challenged.

In the future work, we will solve imperfection of our method; matching ambiguously image, and improve efficiency of the algorithm. We will develop our method for color image.



**Figure 14: Points mapped from source to destination shifted by 10 pixels in x axis**



**Figure 15: Points mapped from source to destination image reduce size 20 % of source image**

## 8. REFERENCES

- [Bdm81a] B.D. McKay, Practical Graph Isomorphism, *Congressus Numerantium*, 30, pp. 45-87, 1981.
- [BJNS03a] B.Chambers, J.Durand, N.Gans, and Seth Hutchinson, Dynamic Feature Point Detection for Visual Servoing, *IROIS 2003*, vol 1, pp. 504-509, Oct. 2003
- [Chr96a] Christian Heipke, "Overview of Image Matching Techniques" Retrieved October 1, 2005, from [http://phot.epfl.ch/workshop/wks96/art\\_3\\_1.html](http://phot.epfl.ch/workshop/wks96/art_3_1.html)
- [Fra01a] Franz Rottensteiner, "Image pyramids" Retrieved October 30, 2005, from <http://www.ipf.tuwien.ac.at/fr/buildings/diss/node105.html>

[Jcu02a] J.Curand, Real-time object tracking using multi-resolution critical point filters, Master's thesis, University of Illinois at Urbana-Champaign, Dec. 2002.

[Jcu03b] J.Curand, Real-time object tracking using multi-resolution critical point filters, *IEEE Int.Conf Robots and Automation(ICRA'03)* , vol.2, pp.1682-1687, Sept. 2003

[Jdu02c] J.Durand, Real-time object tracking using multi-resolution critical point filters, Master's thesis, University of Illinois at Urbana-Champaign, Dec. 2002.

[PAD02a] P.Mohanta ,A.Sen ,D.Mukherjee, Segmentation of Images using Level Set Analysis, *ICVGIP 2002*, vol 22, Dec. 2002.

[RR02a] Rafael C. Gonzalez and Richard E. Woods. **Digital Image Processing**. 2nd ed. [New Jersey]: prentice Hall, c2002.

[YT98a] Y.Shinagawa and T.L. Kunii, Unconstrained Automatic Image Matching Using Multiresolutional Critical-Point Filter, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp.994-1010, Sept.1998.

# Automatic Identification of Ambiguous Prostate Capsule Boundary Lines Using Shape Information and Least Squares Curve Fitting Technique

Rania Hussein  
DigiPen Institute of Technology  
Department of Computer Engineering  
Redmond, WA, 98052  
[rhussein@digipen.edu](mailto:rhussein@digipen.edu)

Frederic D. McKenzie  
Old Dominion University  
Department of Electrical and Computer  
Engineering  
Norfolk, VA, 23529  
[fmckenzi@ece.odu.edu](mailto:fmckenzi@ece.odu.edu)

## ABSTRACT

Currently there are few parameters that are used to compare the efficiency of different methods of cancerous prostate surgical removal. An accurate assessment of the percentage and depth of extra-capsular soft tissue removed with the prostate by the various surgical techniques can help surgeons with determining the appropriateness of different surgical approaches. Additionally, an objective assessment can allow a particular surgeon to compare individual performance against a standard. In order to facilitate 3D reconstruction and objective analysis and thus provide more accurate quantitation results when analyzing specimens, it is essential to automatically identify the capsule boundary that separates the prostate gland tissue from its extra-capsular tissue. However the prostate capsule is sometimes unrecognizable due to the naturally occurring intrusion of muscle and connective tissue into the prostate gland. At these regions where the capsule disappears, its contour can be arbitrarily reconstructed with the generation of a continuing contour line based on the natural shape of the prostate gland. We present an algorithm based on a least squares curve fitting technique that uses a prostate shape equation to merge previously detected capsule parts with the shape equation to produce an approximated curve that represents the prostate capsule. We have tested our algorithms using three shapes on 13 prostate slices that are cut at different locations from the apex and the results are promising.

## Keywords

prostate capsule, least squares, shape information, automatic detection

## 1. INTRODUCTION

Despite the numerous research studies in segmenting structures from medical images [Aar94, Liu97, Pra92, Ric96, and Pat00] and reconstructing a compact geometric representation of these structures, no study, to the best of our knowledge, has been done to automatically identify the complete prostate capsule in medical images.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright UNION Agency – Science Press, Plzen, Czech Republic.

As studies show, identifying the prostate capsule is essential in staging prostate cancer and it greatly affects the treatment options since the presence of metastases in the prostate's adjacent organs is highly related to the penetration through the prostate capsule, which therefore influences the prognosis after surgical and hormonal treatment [McN90]. In addition to its importance in prostate prognosis, automatically identifying the prostate capsule provides a more accurate and objective assessment of the percentage and depth of extra-capsular soft tissue removed with the prostate by the various surgical approaches. Not only does this assessment allow surgeons to compare the quality of one surgical approach versus another, it also provides an evaluation of surgeons' surgical performances as related to a standard [McK03]. Recent studies are focused more on a statistical model based segmentation algorithms [Pra92, Lor97, Gon04, Chi04, Bet05] than deformable models [Kas87, Pat98, Lad00, Kno99a, Kno99b].

By analyzing the existing literature, the error resulting from applying prostate segmentation methods may increase considerably when the image contains shadows with similar gray level and texture attached to the prostate, and/or when boundary segments are missing. Another obstacle that faces segmentation is the lack of sufficient number of training (gold) samples if a learning technique is used. Although algorithms based on active contours have been used successfully, their major drawback is that they depend on user interaction to determine the initial contour.

Therefore, a new segmentation approach should ideally possess certain properties:

- User interaction (e.g. defining seed points or manually placing initial contour) should be eliminated due to its drawbacks such as time consumption, human bias and/or error.
- Sample-based learning should be avoided because it is difficult to provide a large number of training samples in medical environments.
- Robustness of the segmentation algorithm with respect to the presence of noise and shadow is crucial.
- Shape information should be incorporated into segmentation algorithms to be able to estimate contour segments that are missing in some areas.

Our objective is to establish an algorithm which attempts to avoid the problems that exist in literature and to satisfy the above conditions as much as possible.

## **2. DETECTING PARENCHYMAL CONTOURS AND AREAS OF CAPSULE TISSUE**

In pathology, tissue samples are processed and made into stained tissue sections to be mounted on glass slides for interpretation. Representing the histological features of a slide in digital formats may require using a very high resolution capturing device that can capture the details of the tissue as seen under a microscope.

There are several successful attempts in literature to classify tissues of histopathologic images using texture analysis and image morphology [Dia04, Ham97, Ned00, and Pet04]. For example, the authors of [Pet04] were able to correctly identify the different tissue structures in H&E stained histology slides with  $89\% \pm 0.8$  accuracy. The authors were able to identify fat cells, stroma, nuclei of cells of epithelial origin, and other two types of nuclei that represent inflammatory cells and cancer cells. The slides were scanned using a 40X magnification lens, covering almost all the tissue. Accordingly, we conclude that the collagenous fibers within the prostate capsule as well as the epithelial cells can be

automatically identified using either of the techniques mentioned in Dia04] and [Pet04], given that the slides are scanned at 40X magnification. By detecting the epithelial cells, a parenchymal contour can therefore be generated. As for the tissue parts where the prostate capsule exists, the algorithms mentioned above can automatically detect them and mark them on the slices' images to be used as inputs to our algorithms which we will explain in the following sections.

## **3. ESTIMATION OF THE PROSTATE CAPSULE USING SHAPE INFORMATION**

In order to automatically identify the capsule of the prostate and replace the arduous and costly manual process of detecting it, a software algorithm was developed that recognizes the capsule utilizing elements of prostate anatomy and shape. Certain anatomical features make capsule generally detectable; however, the capsule is unrecognizable in some areas because of the naturally occurring intrusion of muscle into the prostate gland at the anterior apex and fusion of extraprostatic connective tissue with the prostate gland at its base. At these regions where the prostate capsule disappears, its contours need to be reproduced by drawing a continuing contour line from those areas where the capsule can be objectively recognized. The elastic fibers within the prostate capsule can be clearly recognized under the microscope and also under high resolution of scanned digital images. In order to correctly locate those lines, it is essential to detect the parenchymal outer contour of the prostatic glandular epithelial elements [McK03], since the capsule is normally located between this contour and the perimeter of the slice.

### **3.1 A Mathematical Model for a Standard Prostate Shape Top Down Anterior To Posterior**

In general, any prostate has a standard shape that can be defined in terms of equations as we had reported in a previous publication [Hus04]. Those equations can be approximated by the Limaçon curve equation

$$r = b + a \cos \theta$$

The limaçon serves only as an approximation since there is always a degree of roundness to this shape. Therefore, the limaçon serves as a better shape than a circle but may not be as good as an ellipse where more elongated prostate shapes are observed.

### 3.2 Approach

We present a general process that utilizes different shape algorithms to detect the prostate capsule. This process can be summarized as follows:

- Using digital images of prostate slices scanned with 40x magnification, identify the input sections of the prostate capsule (the outer perimeter, the parenchymal contour, and the observable portions of the capsule) automatically using texture analysis techniques [Dia04].
- Use the least squares shape algorithm to generate a curve that interpolates between the parts generated in step 1.
- Adjust the generated curve so that it does not violate any constraints. In our case the constraint is that the curve should be between the parenchymal contour and the prostate perimeter.
- Repeat steps 2 and 3 until a satisfactory threshold is acquired.

#### 3.2.1 The Least Squares Shape Algorithm

The least squares method is a very popular technique used to compute estimations of parameters and to find the best fitting model for discrete data. It is widely used in literature to fit a function (which may represent a certain shape) to a set of data which can be used in many applications including medical imaging [Pi196].

Assuming that we have a number  $n$  of discrete data  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$  and  $f(x)$  is a function for fitting a curve. Therefore,  $f(x)$  has the deviation (error)  $d$  from each data point, i.e.  $d_1 = y_1 - f(x_1)$ ,  $d_2 = y_2 - f(x_2)$ ,  $\dots, d_n = y_n - f(x_n)$

The best-fit curve is the curve that has the minimal sum of the deviations squared from a given set of data, i.e. it is the curve that satisfies the following equation

$$\text{Minimum Least square error } (E) = d_1^2 + d_2^2 + \dots + d_n^2$$

$$= \sum_{i=1}^n d_i^2 = \sum_{i=1}^n [y_i - f(x_i)]^2$$

#### 3.2.2 Implementing the least squares using the limaçon shape equation

We have used the least square error to find the closest location of the prostate shape equation with respect to the parts of the capsule that are present in the tissue (Figure 1)

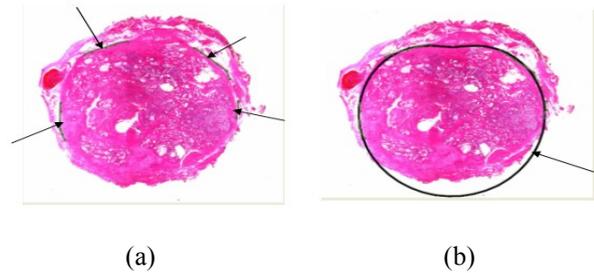


Figure 1. Least squares method and prostate shape equation. (a) Arrows point to the detected parts of the prostate capsule, (b) Arrow points to the curve representing the prostate shape located as close as possible to the capsule parts.

#### Known Capsule Regions Preservation

Once the curve is positioned close to the capsule parts, parts of the shape curve is replaced by the capsule segments and a new curve is generated by connecting all the curve points and capsule points using cubic splines (Figure 2)

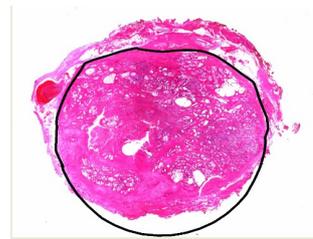


Figure 2. New shape curve after merging the capsule parts into the original shape curve.

#### Curve Adjustment Algorithm

Sometimes the generated curve violates the constraint that states that the prostate capsule is typically located between the parenchymal contour and the prostate perimeter as shown in Figure 3.

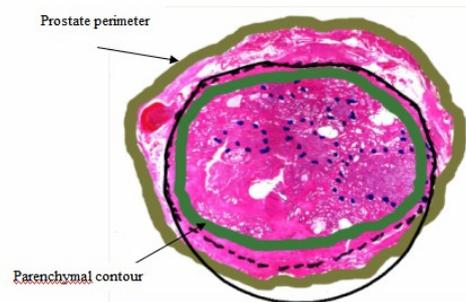


Figure 3. Shape curve extending beyond the prostate perimeter.

In this case, we use the flood fill algorithm to relocate the curve parts that violates the constraint such that new points are generated between the 2

contours (Figure 4) for the least square algorithm to be executed again for better results.

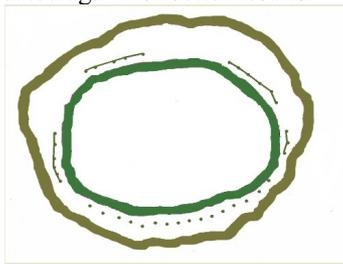


Figure 4. Contours.

The curve adjustment algorithm was used primarily to generate new boundary points to feed the shape algorithm for consecutive runs for improved shape fitting. However, in case that one wishes to stop after a certain number of runs and the output curve extends beyond the slice perimeter or inside the parenchymal contour, the algorithm is used as a final step to enforce this constraint. This enforcement may result in some sharp edges; a curve smoothing technique can be added as a future extension to our algorithm to solve this problem.

#### 4. PERFORMANCE EVALUATION

To evaluate the performance of the least squares algorithm, we have used two measurements, the root mean square error RMSE and the percentage of error, which are defined as follows:

##### Root Mean Square Error (RMSE):

Assuming that curves are represented by control points, the mean square error is the average of squared deviations. Deviations can be calculated by getting the distance from each point on the curve to the closest point on the reference curve. The root mean square error can be calculated by getting the square root of the mean square error as shown in the following equation:

$$RMS = \sqrt{\frac{\sum_{i=1}^n d_i^2}{n}}$$

Where

$n$  is the number of points in the curve

$d_i$  is the min distance from point  $i$  in the curve to the reference curve. The following figure shows the RMS error of the least squares algorithm

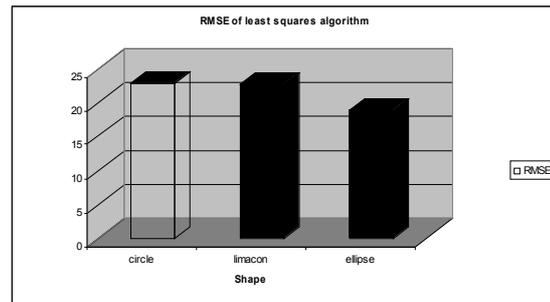


Figure 5. Root Mean Square Error for the least squares algorithm.

##### Percentage Error:

$$\text{Percentage Error} = \frac{\sum_{i=1}^m t_i}{m}$$

Where

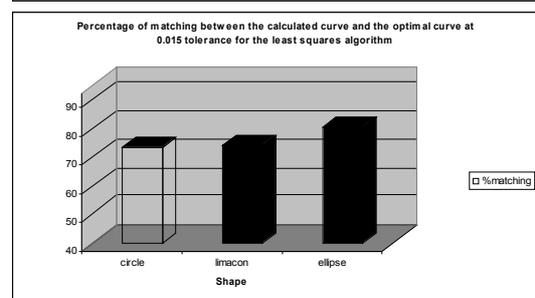
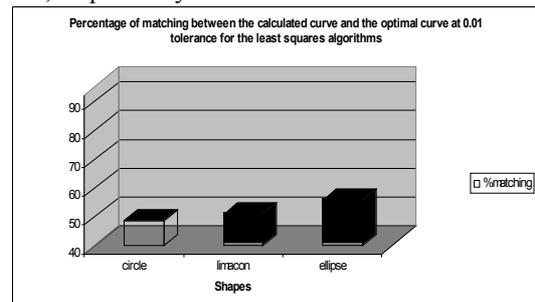
$m$  is the number of points in the reference curve

$$t_i = \begin{cases} 1 & d_i > \text{threshold} \\ 0 & d_i \leq \text{threshold} \end{cases}$$

$d_i$  is the min distance from point  $i$  in the reference curve to the curve

The thresholds considered in our study are equal to 1%, 1.5%, and 2% of the number of pixels of the image diagonal. We found that the 2% threshold, which is the biggest threshold we used, is less than 2mm in length. According to the fact that the capsule thickness is between 0.5 to 2mm [Sat95], we believe that the 2% threshold is reasonable and within acceptable limits while the 1.5% threshold is used to gauge performance improvement. The 1% result is essentially directly on top of the reference line.

The following figures show the % matching between the calculated curve and the optimal curve for the least squares algorithm at 0.01, 0.015 and 0.02 threshold, respectively.



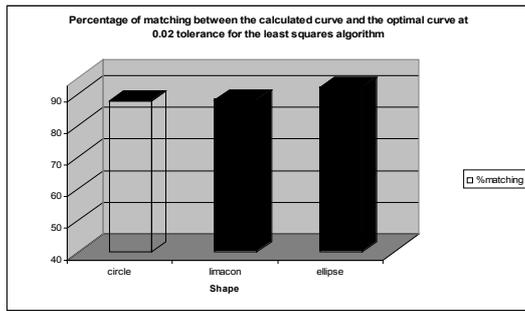


Figure 6. Percentage matching for least squares algorithm.

The results presented show the least squares shape algorithm shows an aptitude for increasing capsule detection as better shape equations are used. The results presented are the outcome of running the algorithm for 2 runs only; however, it can be run for as many times as needed until a satisfactory threshold is acquired. Obviously, increasing the number of runs for a particular specimen is more important for complex prostate equations that have more degrees of freedom.

## 5. CONCLUSION

In this paper, we presented an overall process and a novel shape algorithm to detect the prostate capsule boundary with the use of least squares fitting along with prostate shape equations. We have tested our algorithms on a data set of 13 different prostate slices and our results show promises. Our algorithm show an aptitude for increasing capsule detection as better shape equations are used.

## 6. REFERENCES

- [Aar94] R. G. Aarnik, R. J. B. Giesen, A. L. Huynen, J. J. M. C. H. De La Rosete, F.M.J. Debruyne, and H. Wijkstra, "A practical clinical method for contour determination in ultrasonographic prostate images," *Ultrasound Med Biol.* vol.20, no.7, pp. 705-717, 1994.
- [Bet05] N. Betrounia, M. Vermandela, D. Pasquierc, S. Maoucheb, and J. Rousseaua, "Segmentation of abdominal ultrasound images of the prostate using a priori information and an adapted noise filter," *Computerized Medical Imaging and Graphics*, vol. 29, pp. 43–51, 2005.
- [Chi04] B. Chiu, G.H. Freeman, M.M. Salama, A. Fenster, "Prostate segmentation algorithm using dyadic wavelet transform and discrete dynamic contour," *Phys. Med. Biol.*, vol.49, pp.4943–4960, 2004.
- [Gon04] L. Gong, S. Pathak, D. Haynor, P. Cho, and Y. Kim, "Parametric Shape Modeling Using Deformable Superellipses for Prostate Segmentation," *IEEE Transactions on Medical Imaging*, vol. 23, pp. 340-349, 2004
- [Dia04] J.Diamond, H. Anderson, P. Bartels, R. Montironi, and P. Hamilton, "The use of morphological characteristics and texture analysis in the identification of tissue composition in prostatic neoplasia," *Human Pathology* vol. 35, no. 9, pp. 1121-1131, September 2004.
- [Ham97] P.W. Hamilton, P.H. Bartels, and D. Thompson, "Automated location of dysplastic fields in colorectal histology using image texture analysis," *J. Pathol.*, vol.182, pp. 68–75, 1997.
- [Ham94] P.W. Hamilton, P.H. Bartels, and R. Montironi, "Automated histometry in quantitative prostate pathology," *Anal. Quant. Cytol. Histol.* vol. 20, pp. 443–460, 1998.
- [Hus04] R. Hussein, F. McKenzie, R. Joshi, "Automating prostate capsule contour estimation for 3D model reconstruction using shape and histological features," *SPIE-Int. Soc. Opt. Eng. Proceedings of SPIE* vol. 5367, no.1, pp.790-798, 25 May 2004, USA.
- [Kas87] M. Kass, A. Witkin, D. Terzopoulos, "Snakes: active contour models," *Int. J. Comput. Vision*, vol. 1, pp. 321-331, 1987.
- [Kno99a] C. Knoll, M. Alcaniz, V. Grau, C. Monserrat and M. Juan, "Outlining of the prostate using snakes with shape restrictions based on the wavelet transform," *Pattern Recogn.* vol. 32, no. 10, pp. 1767–1781, 1999.
- [Kno99b] C. Knoll, M. Alcaniz, C. Monserrat, V. Grau and M.C. Juan, "Multiresolution segmentation of medical images using shape-restricted snakes," *Proc. SPIE*, vol. 3661, pp. 222–233, 1999.
- [Lad00] H.M. Ladak, F. Mao, Y. Wang, D.B. Downey, D.A. Steinman and A. Fenster, "Prostate segmentation from 2D ultrasound images," *Med. Phys.*, vol. 27, pp. 1777–1788, 2000.
- [Liu97] Y.J. Liu, W.S. Ng, M.Y. Teo, and H.C. Lim, "Computerised prostate boundary estimation of ultrasound images using radial bas-relief method," *Med. Biol. Eng. Comput.*, vol.35, no.5, pp.445-454, 1997.
- [Lor97] Lorenz, C. Haas and H. Ermert, "Segmentation of ultrasonic prostate images using a probabilistic model based on markov random processes," *Ultrason. Imaging*, vol. 19, pp. 44–45, 1997.
- [McK03] F. D. McKenzie, R. Hussein, J. Seevinck, P. Schellhammer, and J. Diaz. "Prostate Gland and Extra-Capsular Tissue 3D Reconstruction and Measurement," *The 3rd IEEE symposium on Bioinformatics and Bioengineering (BIBE)*, Bethesda, Maryland, pp. 246-250, March 10-12, 2003.
- [McN90] J.E. McNeal, A. Villers, E. Redwine, F. Freiha, and T.A. Stamey, "Capsular penetration in prostate cancer: significance for natural history and treatment," *Am. J. Surg. Pathol.*, vol. 14, pp. 240-247, 1990.
- [Ned00] P. Nedzved, "Morphological Segmentation of Histology Cell Images," *IEEE ICPR'00*, vol.1, pp.1500, 2000.
- [Pat98] S. Pathak, R. Aarnink, J. de la Rosette, V. Chalana, H. Wijkstra, F. Debruyne and Y. Kim, "Quantitative three-dimensional transrectal ultrasound for prostate imaging," *Proc. SPIE*, vol. 3335, pp. 83–92, 1998.
- [Pat00] S. Pathak, D. Haynor, and Y. Kim, "Edge-guided boundary delineation in prostate ultrasound images," *IEEE Transactions on Medical Imaging*, vol.19, no.12, pp.1211-19. Publisher: IEEE, USA, Dec. 2000.
- [Pet04] S. Petushi, C. Katsinis, C. Coward, F. Garcia, and A. Tozeren, "Automated identification of microstructures on histology slides," *2nd IEEE International Symposium on Biomedical Imaging: Macro to Nano (IEEE Cat No. 04EX821)*. IEEE, vol. 1, pp. 424-471. Piscataway, NJ, USA, 2004.

- [Pil96] M. Pilu, A.W. Fitzgibbon, R.B. Fisher, "Ellipse-specific direct least-square fitting," *Proceedings. International Conference on Image Processing (Cat. No.96CH35919). IEEE.* vol.3, pp. 599-602, New York, NY, USA, 1996.
- [Pra92] J.S. [Prater](#), W.D. [Richard](#), "Segmenting ultrasound images of the prostate using neural networks," *Ultrason. Imag.*, vol. 14, pp. 159-185, 1992.
- [Ric96] W.D. Richard, C.G. Keen, "Automated texture-based segmentation of ultrasound images of the prostate," *Computerized Medical Imaging & Graphics*, vol.20, no.3, pp.131-140. Publisher: Elsevier, UK, May-June 1996.
- [Sat95] A. Sattar, J. Noël, J. Vanderhaeghen, C. Schulman, and E. Wespes, "Prostate capsule: computerized morphometric analysis of its components," *Urology*, vol.46, no.2, pp.178-181, 1995.
- [Zhu06] Y. Zhu, S. Williams and R. Zwiggelaar, "Computer technology in detection and staging of prostate carcinoma: A review," *Medical Image Analysis*, vol. 10, no. 2, pp. 178-199, April 2006.

# Fuzzy mathematical morphology and its applications to colour image processing

Antony T. Popov  
St. Kliment Ohridski University of  
Sofia, Bulgaria  
Faculty of Mathematics and  
Informatics  
atpopov@fmi.uni-sofia.bg

## ABSTRACT

In image analysis and pattern recognition fuzzy sets play the role of a good model for segmentation and classifications tasks when the regions and the classes cannot be strictly defined. One of the most widely used fuzzy approaches in image processing is fuzzy mathematical morphology because of the underlying structure of complete lattices. Thus, mathematical morphology is used mainly in processing of binary or single - valued intensity (grey - scale) images for which a partial ordering, hence a lattice structure, is apparent. However, the problem of morphological image processing of colour images is that it is not naturally clear how to define ordering in a colour space. This paper shows a possible way for solving this problem. It is based on the usage of YCrCb colour space as physically intuitive and easy to compute.

**Keywords:** Complete lattice, fuzzy sets, colour models, morphological operations.

## 1 INTRODUCTION

Fuzzy mathematical morphology has been developed to soften the classical binary morphology so as to make the operators less sensitive to image imprecision. It can also be viewed simply as an alternative grey-scale morphological theory. The use of morphological operations require ordering of 3 dimensional colour space, which is hard to be done in natural way, because one should have to order the colours. A way to this is to refer to psychophysiological experiments showing the level of stimulation of eye retina. In our work we use the YCrCb colour model for its simplicity and efficient computation. We obtain adjoint fuzzy erosions and dilations necessary for the construction of efficient filters for colour images denoising without changing the natural colours by subdivision of the CrCb square. This subdivision guarantees the result in particular in the presence of data uncertainties in vision systems.

## 2 MATHEMATICAL MORPHOLOGY – BACKGROUND

Serra [Serra88] and Heijmans [Heijmans94] have shown that morphological operations can be formulated on any complete lattice. A set  $\mathcal{L}$  with a partial ordering “ $\leq$ ” is called a *complete lattice* if every subset  $\mathcal{H} \subseteq \mathcal{L}$

has a supremum  $\bigvee \mathcal{H} \in \mathcal{L}$  (least upper bound) and infimum (greatest lower bound)  $\bigwedge \mathcal{H} \in \mathcal{L}$ .

An operator  $\varphi : \mathcal{L} \mapsto \mathcal{M}$ , where  $\mathcal{L}$  and  $\mathcal{M}$  are two complete lattices, is called dilation if it distributes over arbitrary suprema:  $\varphi(\bigvee_{i \in I} X_i) = \bigvee_{i \in I} \varphi(X_i)$ , and erosion if it distributes over arbitrary infima. Erosions and dilations are increasing operations [Heijmans94]. An operator  $\psi : \mathcal{L} \mapsto \mathcal{L}$  is called a closing if it is increasing, idempotent ( $\psi^2 = \psi$ ) and extensive ( $\psi(X) \geq X$ ). An operator  $\psi$  is called an opening if it is increasing, idempotent and anti-extensive ( $\psi(X) \leq X$ ) [Heijmans94]. A pair of operators  $(\varepsilon, \delta)$ ,  $\delta : \mathcal{L} \mapsto \mathcal{M}$ ,  $\varepsilon : \mathcal{M} \mapsto \mathcal{L}$ , is called an adjunction, if for every two elements  $X \in \mathcal{L}, Y \in \mathcal{M}$  it follows that  $\delta(X) \leq Y \iff X \leq \varepsilon(Y)$ . In [Heijmans94] it is proved that if  $(\varepsilon, \delta)$  is an adjunction then  $\varepsilon$  is erosion and  $\delta$  is dilation. If  $(\varepsilon, \delta)$  is an adjunction, then the composition  $\varepsilon\delta$  is a closing in  $\mathcal{L}$ , and  $\delta\varepsilon$  is an opening in the lattice  $\mathcal{M}$ . As an example, let us consider the lattice  $\mathcal{L}$  with elements the subsets of a linear space  $E$ . Then every translation-invariant dilation is represented by the standard Minkowski addition:  $\delta_A(X) = A \oplus X = X \oplus A$ , and its adjoint erosion is given by Minkowski subtraction:  $\varepsilon_A(X) = X \ominus A$  [Heijmans94]. Then closing and opening of  $A$  by  $B$  are defined as  $A \bullet B = (A \oplus B) \ominus B$ ,  $A \circ B = (A \ominus B) \oplus B$ . These operations are referred to as *classical* or *binary morphological operations*. Openings and closings are generally used as filters for denoising of binary images.

## 3 FUZZY MORPHOLOGICAL OPERATIONS

Consider the set  $E$  called the universal set. A fuzzy subset  $A$  of the universal set  $E$  can be considered as a

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright UNION Agency – Science Press,  
Plzen, Czech Republic

function  $\mu_A : E \mapsto [0, 1]$ , called the membership function of  $A$ .  $\mu_A(x)$  is called the degree of membership of the point  $x$  to the set  $A$ . The ordinary subsets of  $E$ , sometimes called 'crisp sets', can be considered as a particular case of a fuzzy set with membership function taking only the values 0 and 1. This definition leads to two possible interpretations:

- in image representation the value of the membership function  $\mu_A(x)$  at a point  $x$  may be interpreted as the grey level value associated with that point of the image plane,

- in pattern recognition, the value  $0 \leq \mu_A(x) \leq 1$  indicates the probability that the point  $x$  is in the foreground of an image.

The usual set-theoretical operations can be defined naturally on fuzzy sets: Union and intersection of a collection of fuzzy sets is defined as supremum, resp. infimum of their membership functions. Also, we say that  $A \subseteq B$  if  $\mu_A(x) \leq \mu_B(x)$  for all  $x \in E$ . The complement of  $A$  is the set  $A^c$  with membership function  $\mu_{A^c}(x) = 1 - \mu_A(x)$  for all  $x \in E$ . Further, for simplicity, if there is no confusion we will write  $A(x)$  instead of  $\mu_A(x)$ .

If the universal set  $E$  is linear, like the  $n$ -dimensional Euclidean vector space  $\mathbf{R}^n$  or the space of integer vectors with length  $n$ , then any geometrical transformation, like scaling, translation etc., of a fuzzy set can be defined by transforming its  $\alpha$ -cuts [Nguyen00].

Further we consider a simple way to generate adjoint fuzzy morphological operations.

Say that the function  $c(x, y) : [0, 1] \times [0, 1] \mapsto [0, 1]$  is conjunctor if  $c$  is increasing in the both arguments,  $c(0, 1) = c(1, 0) = 0$ , and  $c(1, 1) = 1$ .

Say that the function  $i(x, y) : [0, 1] \times [0, 1] \mapsto [0, 1]$  is implicator if  $i$  is increasing in  $y$  and decreasing in  $x$ ,  $i(0, 0) = i(1, 1) = 1$ , and  $i(1, 0) = 0$ .

Say that the conjunctor - implication pair is adjoint if  $c(b, y) \leq x$  is true if and only if  $y \leq i(b, x)$ . Then for fixed  $b$  function  $f(x) = i(b, x)$  is an erosion, and its adjoint dilation is  $g(y) = c(b, y)$ .

Then having an adjoint conjunctor - implicator pair, as proposed in [Heijmans00], we can define an adjoint pair of fuzzy erosion and dilation:

$$\delta_B(A)(x) = \sup_y c(B(x-y), A(y)),$$

$$\varepsilon_B(A)(x) = \inf_y i(B(y-x), A(y)).$$

Heijmans [Heijmans00] has proposed a number of following conjunctor - implicator pairs to construct morphological operations. Here we give examples of two of them:

$$c(b, y) = \min(b, y),$$

$$i(b, x) = \begin{cases} x & x < b, \\ 1 & x \geq b \end{cases}.$$

These operations are known as operations of Gödel-Brouwer.

$$c(b, y) = \max(0, b + y - 1),$$

$$i(b, x) = \min(1, x - b + 1).$$

These operations are suggested by Lukasiewicz.

Most often the first conjunctor - implicator pair is used. The respective dilation has the form

$(\delta_B(A))(x) = \sup_b \min(A(b), B(x-b))$ . In this case we can denote  $\delta_A(B) = \delta_B(A) = A \oplus B$ , because this operation can be obtained also directly from the binary Minkowski addition using the *extension principle*.

Note that in these cases the conjunctor is symmetric, i.e. it is a t-norm [Nguyen00], and therefore we have  $\delta_A(B) = \delta_B(A)$ .

## 4 MORPHOLOGICAL OPERATIONS FOR COLOUR IMAGES

We saw that the basic morphological operations are expressed as products of the suprema and the infima of the lattice under study. When we deal with colour images, we work in fact in a multidimensional space (usually  $\mathbf{R}^3$  or  $\mathbf{Z}^3$ ), where a natural ordering of the elements cannot be achieved. Therefore we try to introduce some heuristics and to compromise with the accuracy at acceptable level to guarantee the lattice properties and therefore to ensure idempotent opening and closing filtering.

A useful implementation of a basic subjective colour model is the HSV (hue, saturation, value) cone [Rogers98] and its slight modification HLS [Hanbury01]. It was created by A. R. Smith in 1978. It is based on such intuitive colour characteristics as tint, shade and tone (or family (hue), purity (saturation) and intensity (value/luminance)). The coordinate system is cylindrical, and the colours are defined inside a cone for the case of HSV and a double cone in the case of HLS. The hue value  $H$  runs from 0 to  $2\pi$ . The saturation  $S$  is the degree of strength or purity and varies from 0 to 1. The saturation shows how much white is added to the colour, so  $S=1$  makes the purest colour (no white). Brightness (value)  $V$  also ranges from 0 to 1, where 0 is the black. We could experience that in general the value/luminance component contains most of the information about the texture and boundary location. The hue component is usually the most homogeneous (with small variance). The saturation component allows to differentiate between different shades of the same colour. The hue is measured as the polar angle in a plane through the saturation segment and orthogonal to the cone axis. This angle is measured counterclockwise from red. Instead of  $[0, 2\pi)$  we may use the interval  $[0, 1)$  dividing the angle by  $2\pi$ . If  $S=0$  the hue is undefined, and therefore the colour is

achromatic, namely a shade of grey. The fully saturated primary colours occur when  $S=1$ .

However, when we use the HSV or HLS model, the main obstacle is the fact that the hue is measured as an angle, and it is not defined for the levels of grey. In [Hanbury01] a circular 'ordering' of the hue modulo  $2\pi$  is defined, however this ordering does not lead to a complete lattice. Therefore in this case the obtained operators provide good results for some segmentation tasks, but the usage of openings and closings for denoising and filtering is risky. In general, there are no clear mathematical reasons for hue ordering. However, from psychophysiological point of view one may order the colours in the following way - red, magenta, blue, yellow, cyan, green, based on the way humans perceive the hue of the colour. Red is considered to be the smallest, since it stimulates the eye less than the other colours. Contrary, green mostly stimulates the eye [Louverdis02]. In [Hanbury02] an interesting approach for creating colour morphology is presented based on  $L^*a^*b^*$  (CIELAB) colour space representation (the two models are related by the fact that  $H = \arctan\left(\frac{b^*}{a^*}\right)$ ). There the authors divide this space into equipotential surfaces. Unfortunately, the best order for the colour vectors in the same equipotential surface is not obvious. In order to obtain a complete ordering of the colour vectors, they make use of the lexicographical order. Therefore in this case we have idempotent closing and opening filters. However, the transformation from RGB to  $L^*a^*b^*$  is non-linear and time consuming. Moreover, there exists no unique inverse transform. The inverse transform depends on the way we characterize the white point. If one knows the illumination conditions used when acquiring the image, then the specification of the white point is simple. However, if the illumination conditions are unknown, a heuristic hypothesis should be made.

An alternative approach is to use the YCrCb colour model [Rogers98]. To obtain the parameters Y, Cr and Cb we use a simple linear combination of R, G, B values. Note, that Y represents the *lightness*, and should not be mistaken with the yellow colour in the RGB model notation. The parameter Cr encodes the red-cyan sensation, with value  $\approx 0$  for the cyan colour and  $\approx 1$  for the red. The parameter Cb encodes the yellow-blue sensation with  $\approx 0$  indicating yellow and  $\approx 1$  indicating blue. Without lack of generality we can assume that R, G and B values are represented as points in an unit cube, namely  $0 \leq R, G, B \leq 1$ . The YCrCb colour space is also a unit cube with transformation formulas:

$$Y = 0.299R + 0.577G + 0.114B, \quad (1)$$

$$Cr = 0.5R - 0.411G - 0.081B + 0.5, \quad (2)$$

$$Cb = 0.5B - 0.169R - 0.325G + 0.5. \quad (3)$$

Henceforth it is clear that the transformation between RGB and YCrCb models is linear and easy to compute. In colour image processing we give priority to the luminance parameters (V in HSV, L in HLS and Y in YCrCb), since if anyone looks at them, he can usually distinguish the different objects on the image as looking on a black -and- white TV. In general, in colour image processing less priority is usually given to the chrominance maps - for instance H and S in HSV model, or Cr and Cb in YCrCb. When working with HSV model, the second priority is given to the hue, because it contains mostly the colour information, and the least priority is given to the saturation because it is correlated with the other two components and its role as a parameter in image processing tasks is sometimes criticized - one can refer for instance to the work [Hanbury01]. This is another reason to prefer the YCrCb model, where the components Cr and Cb have equal weights.

Let us divide the interval  $[0, 1]$  into  $N$  equal pieces  $I_i = \left(\frac{i-1}{N}, \frac{i}{N}\right]$ . ( $0 \in I_1$ ). Let us also suppose that for a pixel  $x$   $Cb(x) \in I_j$  and  $1 - Cr(x) \in I_i$ . Note that we use the negation of  $Cr$  to obtain an ordering closer to the one presented in [Louverdis02]. Thus we code approximately with accuracy  $1/N$  the  $Cr$  and  $Cb$  values by the number of the step at which we visit the respective cell. Let us consider the cell  $(i, j)$  and let  $n = \max(i, j)$  and  $m = \min(i, j)$ . Then for the number of the step we can prove by induction that

$$T(i, j) = \begin{cases} n^2 - m + 1 & (*) \\ n^2 - 2n + m + 1 & (**) \end{cases}$$

The case (\*) means that either  $n$  is even and  $n = i$ , or  $n$  is odd and  $n = j$ . The case (\*\*) is the opposite one. Further on, for simplicity, for any pixel  $x$  we will denote its respective colour integer code by  $T(x)$ . Then if given a colour image  $X$ , we define the transformation

$$(\chi(X))(x) = \frac{N^2[(N^2 - 1)Y(x)] + T(x) - 1}{N^4 - 1},$$

which is a real number between 0 and 1. Then it is clear that having  $\chi(X)$ , we can find  $Y(x)$  with accuracy  $1/N^2$  and  $Cr(X)$  and  $Cb(x)$  with  $1/N$  simply by taking the quotient and the remainder of the division of  $[\chi(X)(N^4 - 1)]$  by  $N^2$ . Here  $[t]$  means the integer part of  $t$ . The last transformation we denote by  $\chi^{-1}$ . Then it is obvious that  $\chi^{-1}(\chi(X))$  gives an approximation of the original colour image  $X$ , while for any grey-scale image  $\chi(\chi^{-1}(Y)) = Y$ .

Then we could order the colour images, namely say that  $A \prec B$  if  $\chi(A) \leq \chi(B)$ . In this case, if  $A \prec B$  and  $B \prec A$  doesn't mean that  $A = B$ , but means that they are close enough and lie in the same equivalence class. Then we can give correct definition of colour fuzzy morphology, simply when we are given an adjoint conjunctor - implicator pair we can define dilation-erosion adjunction as:

$$\delta_B(X)(x) = \chi^{-1} \left[ \bigvee_y c(B(x-y), (\chi(X))(y)) \right], \quad (4)$$

$$\varepsilon_B(X)(x) = \chi^{-1} \left[ \bigwedge_y i(B(y-x), (\chi(X))(y)) \right]. \quad (5)$$

Here  $B$  can be any fuzzy structuring element, i.e. an image which pixel values are real numbers between 0 and 1.

To compute easily  $\chi^{-1}$  we represent the function  $T(i, j)$  by a  $N^2 \times 2$  table in which the number of the row  $s$  means the current step of the zigzag line, while its columns hold the numbers of the intervals  $i$  and  $j$  such that  $T(i, j) = s$ . In the examples shown in the next section we use value  $N = 16$ .

## 5 EXPERIMENTS AND FURTHER RESEARCH

In this work a new approach for construction of morphological filters for colour images is presented. It is based on interval approximation of the colour space. Thus we meet the requirements for image quality and we control the colour accuracy. The same approximation can be applied also in fuzzy algorithms for image enhancement and binarization.

On figure 1 one can see a colour picture of flowers. Next its dilation, erosion, opening and closing by  $3 \times 3$  flat square structuring element are presented. Note that for flat structuring elements the choice of the t-norm - implicator pair is not essential, which follows easily from the properties of fuzzy t-norms and the uniqueness of the adjoint erosion [Popov00]. The second application of the opening (closing) operation on the opened (closed) image does not affect it due to the idempotence of thus generated opening and closing filters. Thus in our case we check experimentally our theoretical result for the opening and closing idempotence. On figure 2 we show the opening and closing top-hat transforms and the gradient of a picture of a traditional Bulgarian table on Easter holiday. The structuring element  $B$  is a centrally symmetric 'pyramide':

$$B = \begin{pmatrix} 0 & 0.7 & 0 \\ 0.7 & 1 & 0.7 \\ 0 & 0.7 & 0 \end{pmatrix}.$$

Remind that the opening top-hat transform is defined as the difference between the original and the opening, the closing top-hat is the difference between the closing and the original, and the gradient is the difference between the dilation and erosion [Soille02]. However, operations with non-flat structuring elements are rarely used in image filtering, because it is not clear a priori how an incremental operation will affect the

colours. Further experiments with non-flat operators will be made to show the efficiency of our approach in texture analysis. Also we can study the convexity and connectivity of objects on colour images modifying the known approaches for grey - scale images presented in [Bloch93], [Popov97] and [Popov00]. Fuzzy colour morphology operators could be employed in biometric applications including fingerprint identification and removing textured background imprints on bank checks.

## ACKNOWLEDGEMENTS

This work was partially supported by National Ministry of Science and Education under contract BY-TH-202/2006: "Methods and algorithms for analysis of combined biometric information".

## REFERENCES

- [Bloch93] Bloch, I.: Fuzzy connectivity and mathematical morphology, *Pattern Recognition Letters*, **14** (1993) 483-488
- [Hanbury01] Hanbury A. and Serra J.: Mathematical morphology in the HLS colour space, T. Cootes, C. Taylor (eds), *Proceedings of the 12th British Machine Vision Conference - September 2001*, Manchester, UK, Springer (2001) 451-460.
- [Hanbury02] Hanbury A. and Serra J.: Mathematical morphology in the CIELAB Space, *Journal of Image Analysis and Stereology*, **21** (2002) 201-206.
- [Heijmans94] Heijmans, H. J. A. M.: *Morphological image operators*, Academic Press, Boston 1994.
- [Heijmans00] Deng, Ting-Quan, Heijmans, H. J. A. M.: Grey-scale morphology based on fuzzy logic, CWI Report PNA-R0012, Amsterdam, October 2000.
- [Louverdis02] Louverdis G., Andreadis I., and Tsalides Ph.: New fuzzy model for morphological colour image processing, *IEE Proceedings - Vision, Image and Signal Processing*, **149(3)** (2002) 129-139.
- [Nguyen00] Nguyen H.T. and Walker E. A. : *A first course in fuzzy logic* (2nd edition) CRC Press, Boca Raton FL, 2000.
- [Popov97] Popov, A. T.: Convexity indicators based on fuzzy morphology, *Pattern Recognition Letters*, **18 (3)** (1997) 259 - 267.
- [Popov00] Popov, A. T.: Aproximate connectivity and mathematical morphology, J. Goutsias, L. Vincent, D. S. Bloomberg (eds), *Mathematical Morphology and its Applications to Image and Signal Processing*, Kluwer (2000) 149-158.
- [Rogers98] Rogers D. F.: *Procedural elements for computer graphics* (2nd edition), WCB McGraw - Hill, 1998.
- [Serra88] Serra, J.: *Mathematical morphology for complete lattices*. In J. Serra, editor, *Image analysis and mathematical morphology*, vol. 2, Academic Press, London 1988.
- [Soille02] Soille P.: *Morphological image analysis* (2nd edition), Springer-Verlag, Berlin 2002.



Figure 1: From top to bottom and left to right: original, dilation, erosion; opening, closing

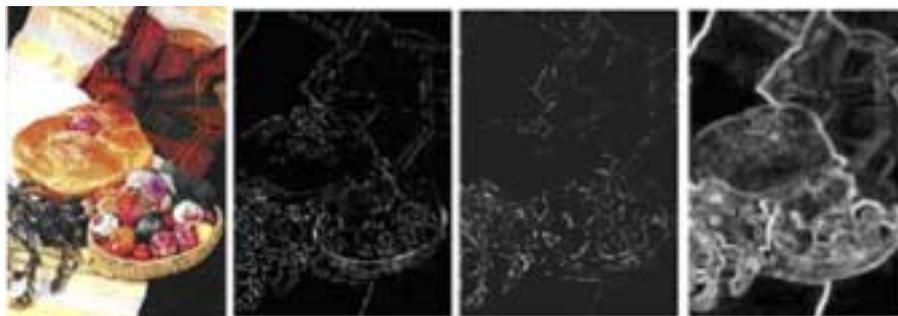


Figure 2: The original colour image, its opening top-hat, closing top-hat and morphological gradient



# Contributions to Colon Segmentation Without Previous Preparation in Computer Tomography Images

Darwin MARTINEZ  
Los Andes University -IMAGINE  
Colombia, Bogota, D.C.  
dar-mart@uniandes.edu.co

José Tiberio HERNANDEZ  
Los Andes University -IMAGINE  
Colombia, Bogota, D.C.  
jhernand@uniandes.edu.co

Leonardo FLOREZ  
INSA-Lyon - CREATIS  
France, Lyon  
leonardo.florez@gmail.com

## ABSTRACT

We propose a method to segment the lumen of the colon from computed tomography (CT) images. To do so, we use first and second order statistical moments. These moments provide us with a set of descriptors to characterize the homogeneity of regions inside the colon.

The algorithm presented in this paper makes use of these values in a prediction-correction exploration process of the colonic region. We show some segmentation results on real patient images that present high non-homogeneous colonic regions.

## Keywords

Segmentation, virtual colonoscopy, image processing, statistical moments, variance, image descriptors, region growing, prediction-correction.

## 1. INTRODUCTION

Colorectal cancer is one major cause of death in the western world [6,7,13]. This disease is less risky if the polyps that cause it are detected in early stages [3,5,7,12,13,18,20,28]. Virtual Colonoscopy (VC), a digital method for polyp detection, is widely accepted because it is less invasive than optical colonoscopy[21].

VC procedure consists on the acquisition of an air-contrasted Computer Tomography (CT) 3D image. This image is then analyzed to identify possible deformations of the colon wall. Analysis is performed by an expert radiologist who uses computer based diagnostic tools specially conceived for:

- colonic lumen segmentation,
- colon central axis computation, and
- polyp detection.

Firstly, the segmentation process is implemented as a threshold filter based on the CT Hounsfield Units

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright UNION Agency – Science Press, Plzen, Czech Republic.

(HU) [11]. These units represent different structures in a standardized scale: air has a value between -1000 HU and -800 HU [5,20,29]. The main problem of a simple threshold is the presence of adjacent organs in the abdominal region: lungs, stomach and small intestine are also filled with air, thus faking the segmentation results. In such cases it would be necessary to use more elaborated strategies to extract the colon and eliminate adjacent structures. The most common strategies are often based on region growing. The user provides a seed set inside the colon and the growing algorithm is adapted to avoid adjacent regions. Some examples of these procedures can be found in [1,2,4,5,12,14-17,20,23,28-30].

Secondly, axis computation is often based on morphological erosion [22] or on the analysis of a distance map [11,28]. Such geometrical object is used as a navigation path, inside the colonic lumen, to guide detection of polyps [22,28].

Finally, detection of polyps is not an easy task. Most methods use a hybrid approach to coarsely detect polyps and then refine this detection. Subsequently, specific rules are applied in order to classify detected regions as polyps.

Some techniques used for polyp identification are:

1. Geometric characterization: different properties such as elliptic curvature, mean curvature, minimum size [1,25], diameter, sphericity [26,27], height, radius, mean intensity[14] are measured. These measurements are affected by the colon distention, resulting from air insufflations.

2. Density characterization: the uniformity of voxel sets identified as polyp candidates in the previous process is studied [26].
3. Deformable models: starting with a selection of identified polyp candidates, this technique places a seed inside each candidate. The model grows until it reaches the surface of the polyp [30].
4. Optic flow: this process evaluates the differences between the edges of polyp candidates (identified by means of geometric characterization in the previous stage) in adjacent slices.

These techniques intend to emulate the method used by the experienced radiologist without computer aid [1,2].

Segmentation is a fundamental part of the process. The quality of polyp detection in the VC procedure depends on the precision of the segmentation stage, both if the detection is performed by the radiologist and if the above mentioned techniques are used.

### Motivation

In order to reduce patient preparation and the invasive characteristics of air contrast VC, we propose to explore the behavior of some segmentation image processing techniques in CT studies of patients with less or no preparation. A particular study of variance

as region descriptor [10,19,24], and the region explorers based on the prediction-correction technique[9] was made. It's important to mention that the present work does not have clinical intention and it is just an exploration of an alternative segmentation method.

Our method proposes working over a 3D image whose values are the variance of the intensities in the original image computed in each voxel's local neighborhood. We intend to explore the local homogeneity of the colon content (air and feces matter) as a main criterion in segmentation, and the original data as validation parameters in the region growing process.

The method is an interactive process of prediction-correction, using an advance strategy based on an initial vector provided by the user, and a prediction instrument (explorer's set) which guides the process along the colon.

Results show a good behavior both of the variance as workspace for this kind of segmentation, and of the advance strategy along the colon.

### 2. METHODOLOGY

The selected segmentation strategy is composed by two stages:

- Preprocessing and Initialization
- Iterative Segmentation

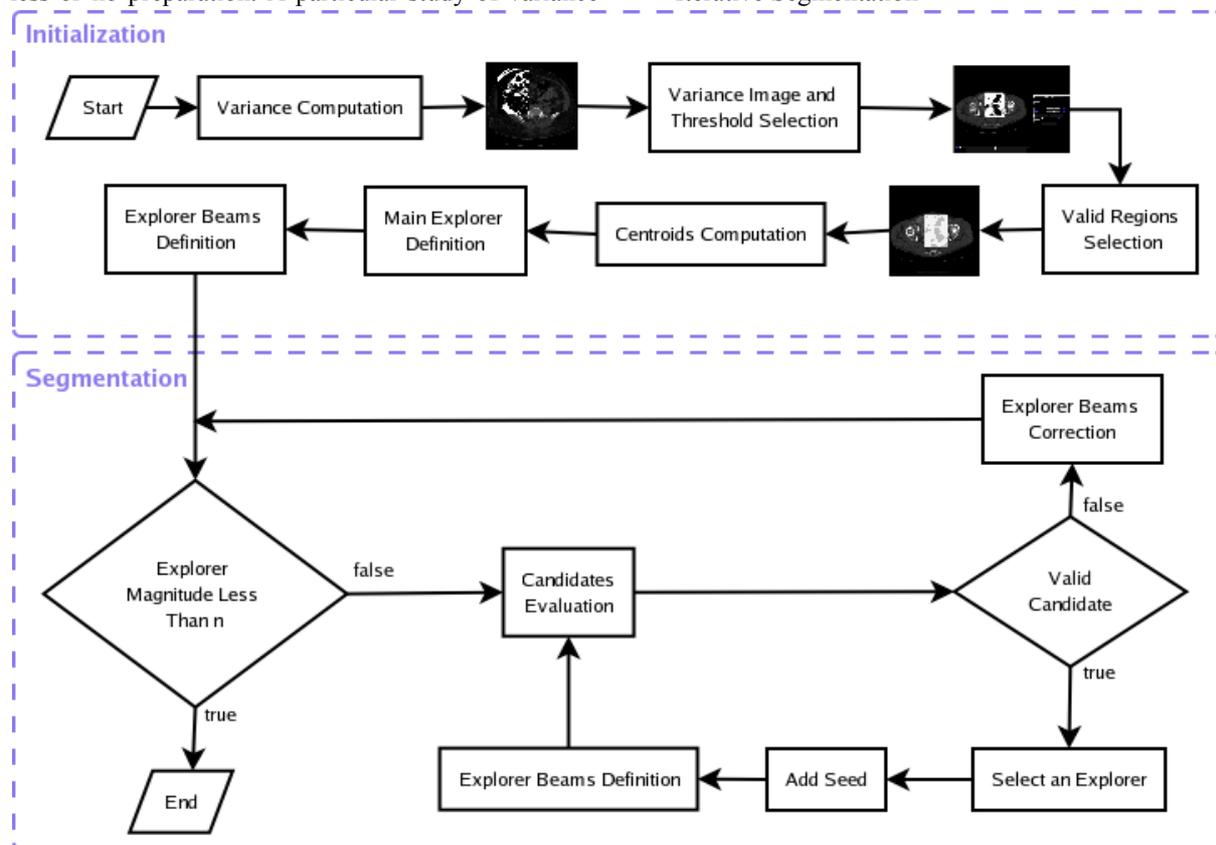
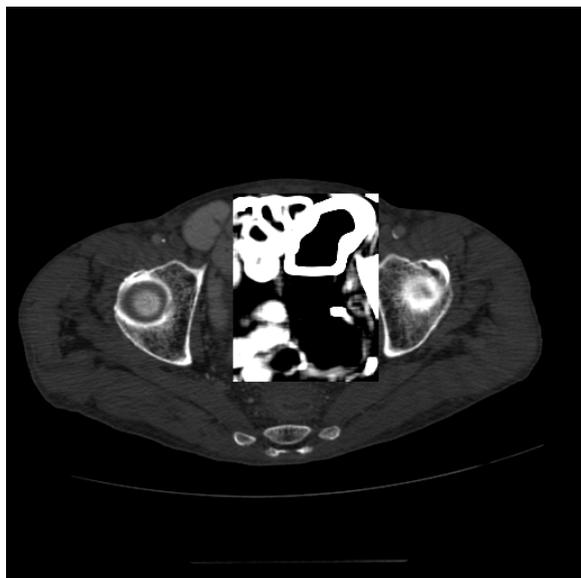


Figure 1: Procedural Flow Diagram

In the preprocessing and initialization stage the *variance image*—starting from the local variance values of the original image—is built using a neighborhood size parameter (between 5 and 11 voxels) defined by the user, who also defines a threshold (in the *variance image*) characterizing the regions inside the colon wall (air and feces matter). The user then defines the first *advance vector*, which must have the origin and end points in the two different valid and adjacent regions.



**Figure 2: Preprocessed Image. Chess representation of variance on original image. Variance was calculated with a cubical kernel of 7 voxels of side.**

The segmentation stage is an iterative process based on the prediction-correction algorithm. In our case, this algorithm uses the *direction vector* of the last iteration as a guide for advance (prediction). An *explorer beam* (EB)—data structure composed by a set of vectors surrounding one main direction and providing associated image information—is used to explore and evaluate the region in order to define a new *direction vector* (correction) and thus launch the growing process. The stop criterion is the failure of the new direction vector search.

## Preprocessing and Initialization

### 2.1.1 Computing the Variance Image

The 3D image preprocessing is based on the computation of the mean and variance values for all the voxels inside the user-selected volume of interest (VOI). This procedure generates two new images, one for the mean and another for the variance values. The selected VOI should contain the two valid regions inside the colon (air and feces matter). The variance values in 3D are computed in the traditional way. It is important, nevertheless, to consider that, when a voxel

neighborhood is not completely contained in the VOI, the formula should be modified to exclude outside voxels.

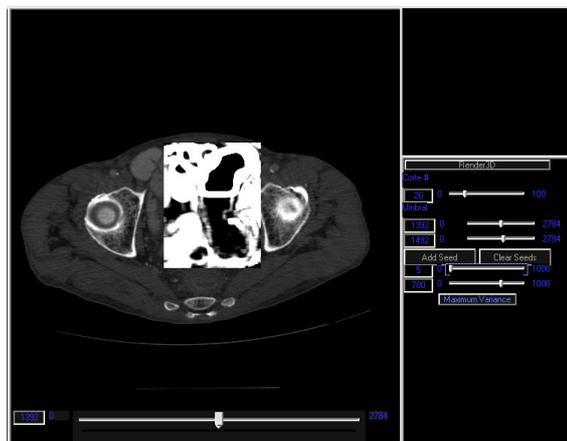
### 2.1.2 Defining Region Growing Threshold

The user selects a threshold on the *variance image*, for which purpose the system offers an interactive visualization of original and variance VOI images. The threshold should be chosen in a way such that in the *variance image* the colon wall (high variance) and the different regions inside the colon (rather low variance) are clearly seen, especially those with feces matter. Also, in some occasions it is difficult to distinguish between the outside of the colon wall and the wall itself.

### 2.1.3 Selecting Valid Regions

As previously mentioned, the procedure requires the identification of two valid and adjacent regions inside the colon, one with feces matter and the other with air. In order to facilitate the selection of valid regions, the system provides an interactive visualization of both the original and the chess variance VOI images.

Each selected region—an interior parallelepiped—is used as a characterization frame of the regions, expressed in the variance and the mean behavior. The behavioral descriptors are the main parameters for both explorer evaluation and region growing steps. These last steps are based on the variance and intensity gray values of each voxel in the image and they define the validity characteristics for the regions.



**Figure 3: Threshold variance manipulation to evidence the differences between regions inside and outside colon. Particularly, it shows the homogeneity of region variance for both kinds of regions inside colon.**

Once a valid two regions selection has been performed, the procedure defines an initial *direction vector*, the first main explorer. This explorer is specified by the magnitude and direction as follows:

1. Computing the regions' centroids,
2. Computing the distance between centroids,

magnitude

3. Defining the direction based on the order of region selection.

The final point of this vector will be the first seed for the region growing, which starts the segmentation process.

#### 2.1.4 Defining the Explorer Beam

The *explorer beam* (EB) is a set of vectors used to guide the advance in the segmentation process. The method creates the EB based on direction and magnitude of the previously selected *direction vector*.

Let us define the main explorer of the EB  $e_p$  as the vector that has the same magnitude and direction of the previous *direction vector*. Its origin is the end point of the previous *direction vector*. Other explorers in the EB are created around  $e_p$  in a way such that a cone with an alpha angle is formed whose vertex is in the origin point. The result is an explorer semi-conic beam. So, the EB would be view as a data structure to search for some characteristics of image as gray value and variance along each vector around  $e_p$ .

In other words, for each new vector in EB, the procedure computes the variance all along, so that a variance profile of each explorer is created. This profile is used in the explorer's evaluation (section 2.2.1).

### Iterative Segmentation

#### 2.2.1 Evaluating the Explorer Beam

The evaluation verifies the existence of a valid vector candidate in the explorer beam, according to the following validity conditions:

1. The variance values associated to the vector voxels are in the valid range defined by the selected valid regions, and
2. The corresponding intensity values are within the valid range defined for the same valid region.

The algorithm searches for explorers that comply with the above conditions. If  $e_p$  is among them, it should be selected. The final point of the selected explorer will become a seed for the region growing stage (section 2.2.4). In the event that no explorer in the current EB fulfills the conditions, the correction step begins (section 2.2.2).

#### 2.2.2 Correcting the Explorer Beam

The objective of this step is to find a complying explorer beam which should contain the next direction vector. The correction uses two different control variables: the magnitude of vectors and the direction of the main explorer (*direction vector*). Even though the conic angle might be used as another control variable, it was not pursued in this study.

When an explorer fails the evaluation step, each of its vectors is labeled with the value of the distance from its origin to the first non-compliant voxel (fail label). The EB correction is calculated from the explorer's fail label distribution, and it proceeds in two different ways:

1. Magnitude correction: the magnitude of all explorers is reduced in half. This correction takes place when the fail labels have similar values.
2. Direction correction: a new main explorer is created by using the explorer with the greater fail label, whose direction will correspond to that of the new main explorer. The new main explorer's magnitude will correspond to the fail label value of the very same explorer. This correction takes place when the fail labels have fairly different values.

This new EB demands a new evaluation process (section 2.2.1).

#### 2.2.3 Stop Criterion

The exploration cycle stops when, after an EB correction, the magnitude of vectors is found to measure less than one unit.

#### 2.2.4 Region Growing

The region growing is started at the end of each iteration. It takes as seed the end of the selected explorer (the new direction vector). This propagation is a recursive method that evaluates the 6-orthogonal neighbors seeking for the voxels that fulfill one of the following two conditions:

1. The estimated variance value for the voxel is in one of the ranges of valid variance,
2. The voxel has an estimated variance smaller than the threshold specified in the initialization stage, and its gray intensity is in the valid range of intensities.

When a voxel fulfills one of the conditions, it is singled out, and the region follows. The algorithm stops when no voxel fulfills at least one condition.

## 3. RESULTS

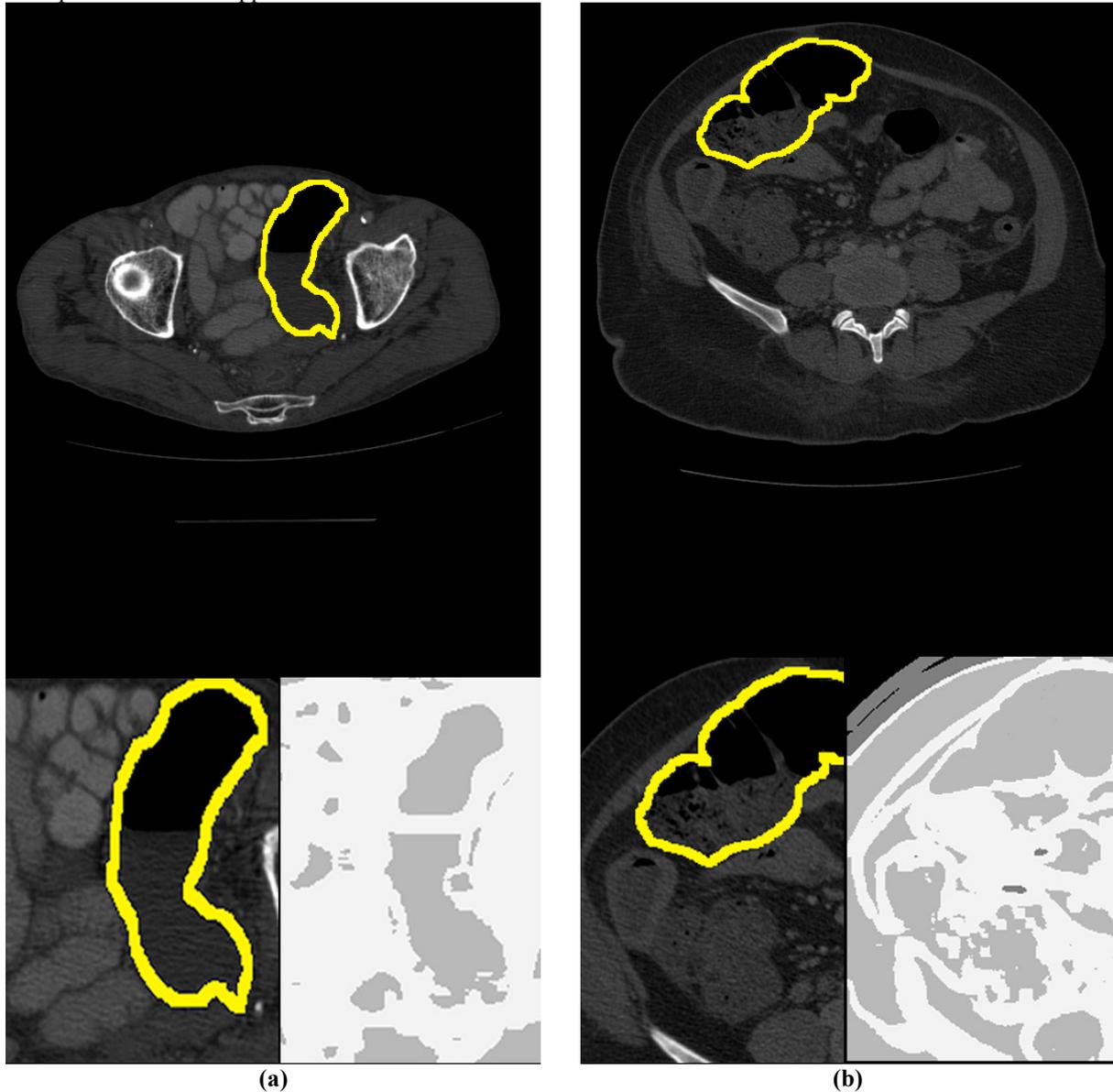
Next, presenting the image homogeneity conditions that shows a good results.

Figure 4 shows two graphics showing the manipulation of variance threshold estimation, which enhances the image characteristics in a chess representation of statistical values on top of the original ones, the a) figure shows an image of the colon with a good homogeneity in the feces matter region, while the b) figure shows an image with high variance in the feces matter region, so the image does not keep the hypothesis of low variance in the regions inside the colon, and for this reason the image were

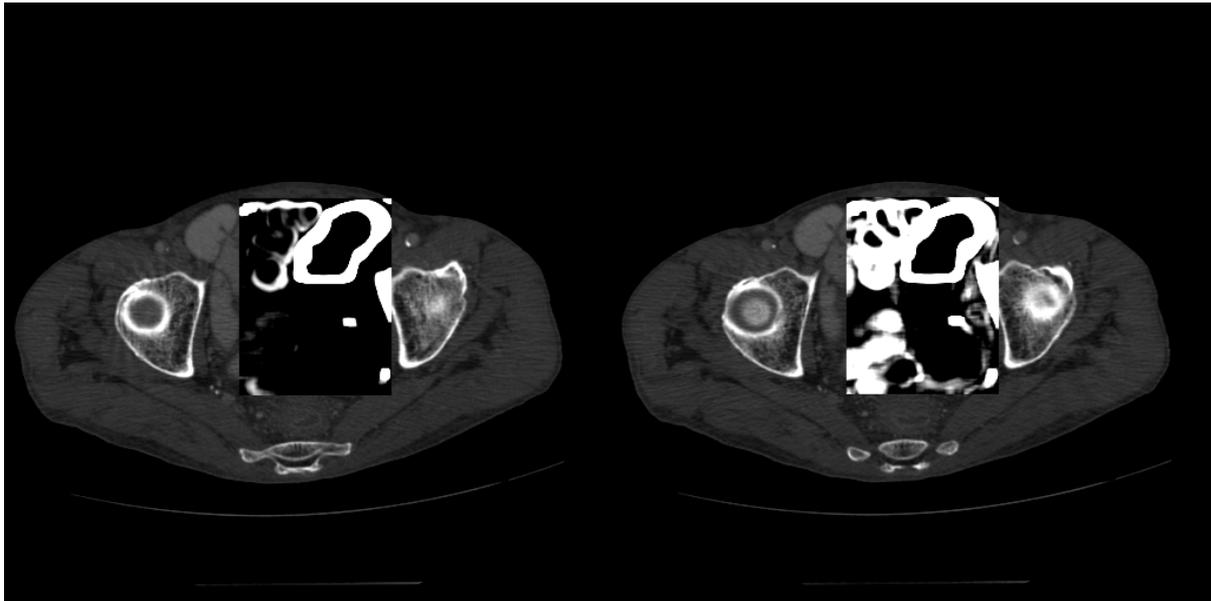
discarded. Figure 5a presents a variance image where the influence of two types of scenarios becomes evident: the transitions between air and organs, and that between organs and organs. Figure 5b illustrates the appearance of a border in the feces matter region inside the colon. This image is obtained through the manipulation of the variance threshold. Finally, figure 6 shows the method's results in a fragmented segmentation of the matter inside the colon.

The procedure was applied to four different CT

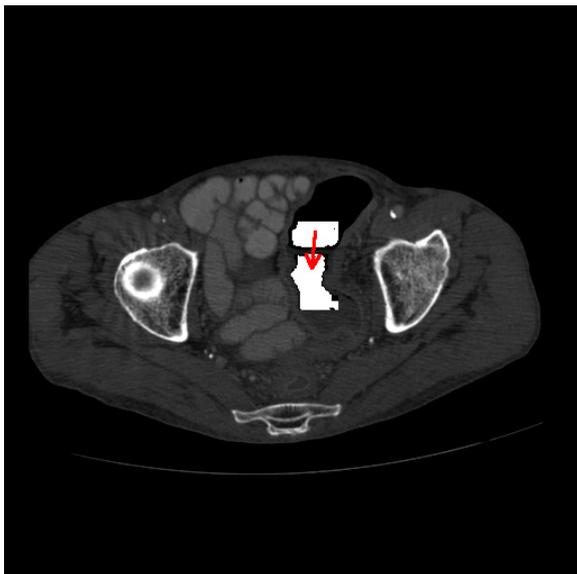
images. These images present different characteristics in region homogeneity inside the colon. All images present an oral contrast medium that lightens the small intestine. Two of these images have homogeneous regions inside the colon with an insufficient size for estimator computation in the initialization stage. In this case, the process did not achieve reliable estimators, and the images were discarded.



**Figure 4: Variance manipulation: the yellow highlighted regions correspond to the colon wall, In both cases they represent the region to segment. a) homogeneous regions inside the colon present low variance b) non homogeneous regions present high variance. It is important to note that image a) fulfills the hypothesis of continuity based on variance while image b) does not.**



**Figure 5: Interactive variance threshold manipulation: The choice of this threshold is critical for the method. In the left side image, a very high threshold hides the relatively low variances of fecal matter, while a lower threshold (right side image) allows the visualization of the internal region of colon (air and fecal matter), surrounded by high variance regions in organs around the colon. In this way, the image of the right side is appropriate for the interactive definition of the two seeds for the segmentation process.**



**Figure 6: A chess representation resulting from the segmentation of regions inside the colon. We can see two white fragments that represent the segmentation in the two valid regions over the original image. The segmentation follows the direction of the red arrow (starting at the first direction vector).**

#### **4. DISCUSSION AND CONCLUSION**

The variance as image regions descriptor gives a good estimation about the region's homogeneity. Statistical

3D computation gives a space continuity condition for the matter. Figure 4 shows two cases where the variance enhances the homogeneity conditions in the regions inside the colon. In a) this condition is particularly notorious, while in b) the region inside the colon shows high variance. In this case, the estimation of regions inside the colon could be misleading because responses to transitions between organs might interfere. This is the reason why these images were discarded. Figure 5 presents two variance images. The first one shows higher variance obtained as a response to changes in the distribution of intensity gray values corresponding to different organs, and to air and osseous structure.

Additionally, the second image indicates how a fragment of the wall is lighter as a response to the influence of the changes in adjacent slices. This determines the threshold condition for the region growing.

In conclusion, we find that the statistical descriptors offer a good estimation of region homogeneity that includes the spatial distribution. Also, the strategy of prediction-correction facilitates the algorithm's easy adaptation to image conditions using local values both to determine the advance direction, and to act as reference values in the region growing process. Additionally, we find an enormous utility in expressing the decision criteria in the region growing process based on more than one characteristic (the variance, mean, and original images) to evaluate the growing conditions.

Based on the proposed sketch, explorer beams evidence a good potential for other applications. A further study of the stop criterion and the correction strategies previously mentioned would be an important development.

## 5. REFERENCES

- [1] B. Acar, C.F. Beaulieu, D.S. Paik, S.B. Göktürk, C. Tomasi, J. Yee, S. Napel, Assessment of an Optical Flow Field-Based Polyp Detector for CT Colonography, in 23rd Annual International Conference of the IEEE Engineering, 2001.
- [2] Burak Acar, Christopher F. Beaulieu, Salih B. Göktürk, Carlo Tomasi, David S. Paik, R. Brooke Jeffrey, Jr., Judy Yee, and Sandy Napel, Edge Displacement Field-Based Classification for Improved Detection of Polyps in CT Colonography, in Medical Imaging IEEE Transactions, 2002.
- [3] American Gastroenterological Association, Cancer Colorectal, <http://www.gastro.org/wmspage.cfm?parm1=686>
- [4] Chirtopher F. Beaulieu, R. Brooke Jeffrey, Chandu Karadi, David S. Paik, Sandy Napel, Display Modes For CT Colonography. Part II: Blinded Comparison Of Axial CT and Virtual Endoscopic and Panoramic Endoscopic Volume Rendered Studies, in RSNA Radiology, 1999.
- [5] Alberto Bert, Ivan Dmitriev, Silvano Agliozzo, Natalia Pietrosevoli, Mario Ferraro, Mark Mandelkern, Teresa Gallo, Daniele Regge, A new automatic method for colon segmentation in CT-Colonography CAD, 2003.
- [6] S. Bond, M. Brady, F. Gleeson, N. Mortesen, Image Analysis for Patient Management in Colorectal Cancer, in CARS, 2005.
- [7] Dongqing Chen, Zhengrong Liang, Mark R. Wax, Lihong Li, Bin Li, Arie E. Kaufman, A Novel Approach to Extract Colon Lumen from CT Images for Virtual Colonoscopy, in IEEE transactions on medical imaging, 2000.
- [8] Ming Wan, Frank Dacheille, Arie Kaufman, Distance-Field Based Skeletons for Virtual Navigation, in IEEE Visualization, 2001.
- [9] Leonardo Flórez, Modèle D'état De Cylindre Généralisé et la quantification de Sténoses Artérielles En Imagerie 3D, 2006.
- [10] Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, 2002.
- [11] Hounsfield Units, [http://www.itk.org/Wiki/ITK\\_Hounsfield\\_Units](http://www.itk.org/Wiki/ITK_Hounsfield_Units)
- [12] Peter W. Hung, David S. Paik, Sandy Napel, Judy Yee, R. Brooke Jeffrey, Andreas Steinauer-Gebauer, Juno Min, Ashwin Jathavedam, Christopher F. Beaulieu, Quantification of Distention in CT Colonography: Development and Validation of Three Computer Algorithms, in RSNA Radiology, 2002.
- [13] Gheorghe Iordanescu, Ronald M. Summers, Benefits of Centerline Analysis for CT Colonography Computer-Aided Polyp Detection, 2003.
- [14] Anna K. Jerebko, Sheldon B. Teerlink, Marek Franaszek, Ronald M. Summers, Polyp Segmentation Method for CT Colonography Computer Aided Detection, in SPIE , 2003.
- [15] Chandu Karadi, Chirtopher F. Beaulieu, R. Brooke Jeffrey, David S. Paik, Sandy Napel, Display Modes For CT Colonography. Part I: Synthesis And Insertion of Polypos into Pacient CT Data, in RSNA Radiology 1999.
- [16] Lihong Li, Dongqing Chen, Sarang Lakare, Kevin Kreeger, Ingmar Bitter, Arie E. Kaufman, Mark R. Wax, Petar M. Djuric, Zhengrong Liang, An image segmentation approach to extract colon lumen through colonic material tagging and hidden Markov random field model for virtual colonoscopy, in CiteSeer 2002.
- [17] Ping Li, Sandy Napel, Burak Acar, David S. Paik, R. Brooke Jeffrey, Christopher F. Beaulieu, Registration of central paths and colonic polyps between supine and prone scans in computed tomography colonography: Pilot study, in NCBI, 2004.
- [18] Wolfgang Luboldt, Peter Bauerfeind, Simon Wildermuth, Borut Marincek, Michael Fried, Jrg F. Debatin, Colonic Masses: Detection with MR Colonography, in RSNA Radiology, 2000.
- [19] R Mukundan, K R Ramakrishnan, Moment Functions In Image Analysis: Theory And Application, 1998.
- [20] David S. Paik, Christopher F. Beaulieu, Geoffrey D. Rubin, Burak Acar, R. Brooke Jeffrey, Jr., Judy Yee, Joyoni Dey, Sandy Napel, Surface Normal Overlap: A Computer-Aided Detection Algorithm With Application to Colonic Polyps and Lung Nodules in Helical CT, in IEEE transactions on medical imaging, 2004.
- [21] Joost F. Peters, Simona E. Griogorescu, Roel Truyen, Frans A. Gerritsen, Ayso H de Vries, Rogier E. van Gelder, Patrik Rogalla. Robust Automated Polyp Detection for low-dose and normaldose virtual colonoscopy, in Elsevier, 2005
- [22] Robert J.T. Sadleir, Paul F. Whelan, Colon Centerline Calculation for CT Colonography Using Optimised 3D Topological Thinning, in CiteSeer, 2002
- [23] Mie Sato, Sarang Lakare, Ming Wan, Arie Kaufman, Zhengrong Liang, mark Wax, An Automatic Colon Segmentation for 3d Virtual Colonoscopy, in IEICE Trans. Information and Systems 2001
- [24] Milan Sonka, Vaclav Hlavac, Roger Boyle, Image Processing, Analysis, and Machine Vision, 1999.

- [25] Ronald M. Summers, Christopher F. Beaulieu, Lynne M. Pusanik, James D Malley, R. Brooke, Jeffrey, Daniel I. Glazer, Sandy Napel, Automated Polyp Detector For CT Colonography: Feasibility Study, in RSNA Radiology, 2000
- [26] Ronald M. Summers, Anna K. Jerebko, Marek Franaszek, James D Malley, Daniel Johnson, Colonic Polyps: Complementary Role Of Computer Aided Detection In CT Colonography, in RNSA Radiology, 2002
- [27] Ronald M. Summers, Daniel Johnson, Lynne M. Pusanik, James D. Malley, Ashraf M. Youssef, Judd E. Reed, Automated Polyp Detection at CT Colonography: Feasibility Assessment in a Human Population. In RSNA Radiology, 2001.
- [28] Ming Wan, Zhengrong Liang, Qi Ke, Lichan Hong, Ingmar Bitter, Arie Kaufman, Automatic Centerline Extraction for Virtual Colonoscopy, in IEEE transactions on medical imaging, 2002
- [29] C.L. Wyatt, Y. Ge, D.J. Vining, Automatic segmentation of the colon for virtual colonoscopy, in Elsevier, 2000.
- [30] Jianhua Yao, Meghan Miller, Marek Franaszek, Ronald M. Summers, Automatic Segmentation of Colonic Polyps in CT Colonography Based on Knowledge-Guided Deformable Models. In SPIE, 2003

# Estimating the tumor-breast volume ratio from mammograms

J. Rodríguez<sup>\*</sup>  
jrodrigu@uc.edu.ve

P. Linares<sup>\*</sup>  
plinares@uc.edu.ve

E. Urra<sup>\*</sup>  
eduardourra@hotmail.com

D. Laya<sup>\*</sup>  
danylaya@hotmail.com

F. Saldivia<sup>\*</sup>  
felipesaldivia@cantv.net

A. Reigosa<sup>♦</sup>  
areigosa@uc.edu.ve

<sup>\*</sup>CEMVICC – Faculty of Science and Technology, University of Carabobo, Valencia, Venezuela

<sup>\*</sup>Breast Pathology Service, Institute of Oncology Dr. Miguel Pérez Carreño, Valencia, Venezuela.

<sup>♦</sup>CIMBUCC – Faculty of Health Sciences, University of Carabobo, Valencia, Venezuela

## ABSTRACT

Each year, 182,000 women are diagnosed with breast cancer and 43,300 die, only in America. Currently, more than a million of new cases will be detected around the world. One woman in eight either has or will develop breast cancer in her lifetime. Mammograms are among the best early detection methods. Most women with breast cancer have some type of surgery: Lumpectomy removes only the breast lump and a surrounding margin of normal tissue. Partial or segmental mastectomy or quadrantectomy removes more breast tissue than a lumpectomy (up to one-quarter of the breast). In a total mastectomy the surgeon removes the entire breast, including the nipple. This operation is sometimes used to treat stage 0 breast cancers. For most women with stage I or II breast cancer, breast conservation therapy (lumpectomy and radiation therapy) is as effective as mastectomy. Survival rates of women treated with these 2 approaches are the same. However, breast conservation therapy is not an option for all women with breast cancer. One of the main factors to decide which type of surgery must be used (lumpectomy, quadrantectomy or total mastectomy) is the tumor-breast volume ratio. Unfortunately, because the mammograms are two-dimensional projections of the breast, a process to measure this relationship with the needed accuracy does not exist. So, a correct decision depends on the specialist's ability to imagine the 3D reconstruction from the mammograms. In order to assist the decision making of the surgeon, in this work we have implemented a software application which allows the surgeon to input the mammograms of the patient and obtain a 3D representation of the breast and the tumor. In summary, the software groups a set of tools to segment the breast and tumor from the input mammograms in an interactive way using the intelligent scissors technique, visualize the mammograms in 3D, reconstruct the 3D representation of the breast and tumor and compute a value which represents the volumetric ratio between the breast's reconstruction and the bounding volume of the tumor. This value can be used as a less subjective indicator to decide whether a lumpectomy or quadrantectomy is advisable.

## Keywords

Breast-Conserving Surgery, Surface Reconstruction, Scientific Visualization .

## 1. INTRODUCTION

Breast Cancer has become a public health problem during the last 50 years. Currently, more than a

million new cases of breast cancer are diagnosed [Par05]. There are numerous breast pathologies, from those that do not involve a surgical intervention to the pathologies that require complex treatments that may include the total extirpation or partial section of the mamma. In this work, interest is focused in the cases where the diagnostic suggests a surgical action and the decision to be taken is whether to carry out a total or partial mastectomy. A main factor to be taken into account by the specialist, to decide what kind of procedure to follow, is based on the appraisal of the volumetric proportion between the tumor and the whole breast. This

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright UNION Agency – Science Press, Plzen, Czech Republic.

proportion is estimated from the patient mammograms, using X-ray projections in two or three orientations, namely, craniocaudal, lateral and medio-lateral oblique [Kop98], and also from the clinician palpation. An acceptable judgment by the specialist demands a broad experience and a special competence to mentally reconstruct the volume from the projection two-dimensional information. In fact, different judgments from different specialists for the same patient occurs frequently. In this work, the design and development of a software application to produce and visualize a volumetric approximation of the breast and the tumor is undertaken. It is also intended to produce a numeric indicator associated to the volume ratio between the tumor and the breast regions so that the specialist can have more objective information when deciding the kind of surgical intervention to be carried out. The software system should be interactive and make use of two digitized mammographic views.

## 2. RELATED WORK

Breast-conserving surgery (BCS), one of the most important innovations in the treatment of breast cancer, started in 1972 as a proposal of Veronesi et al. [Ver77] at the National Cancer Institute of Milan, Italy. Patients were selected for quadrantectomy if breast tumors had a largest diameter of 2 cm or less on physical examination and no palpable axillary nodes [Ver02]. Similarly, Fisher et al. [Fis02] have been working in comparisons with lumpectomy taking into account for selection of patients women with invasive breast tumors that were 4 cm or less in their maximal diameter. Retrospective studies have shown that breast-conserving treatment and mastectomy are equivalent if patients are selected appropriately [Ver02][Fis02]. Contraindications may arise from several factors such as pregnancy, number and location of malignancies, patient medical history, and the size of the tumor in relation to the size of the breast [Win92].

Several works have been previously published in relation to breast cancer and computational applications in order to assist the specialists in the diagnosis and surgical planning. Zwiggelaar et al. [Zwi04] and Petroudi et al. [Pet03] have worked on mammograms in order to detect some structures which can be interpreted as indicators of breast cancer risk. In a technique for both the detection and anatomical classification of linear structures in mammographic images, based on the statistical modelling of scale-orientation signatures, was developed. In [Pet03] an objective quantitative

measure of breast density was proposed based on a new approach to breast parenchymal pattern classification. In order to obtain a good segmentation of the breast from mammograms, Wirth et al. [Wir05] has developed a new algorithm for breast region segmentation using fuzzy reasoning. The algorithm uses morphological pre-processing to suppress artifacts and accentuate the breast region, followed by a fuzzy rule-based algorithm to classify the breast tissue region. Although this technique produces nice results to segment the breast from mammograms, the results for the tumor segmentation is not accurate. Finally, in volume reconstruction of the breast, there are very interesting results with Magnetic Resonance Imaging (MRI)[Cot05][Car04][Sub04], but little work has been done from X-ray mammography. This is true despite X-Ray mammography is the standard in clinical practice and the most widely used method for early detection of breast cancer. Carotenuto et al. [Car04], Subramanian et al. [Sub04] and Coto et al. [Cot05] propose software tools to explore, identify, process, visualize and quantify lesions from breast dynamic contrast enhanced MRI data (DCE-MRI). In this paper a volumetric approximation of the breast and the tumor is undertaken from X-ray mammography data and a numeric indicator associated to the volume ratio between the tumor and the breast regions is computed.

## 3. OVERVIEW OF THE PROPOSAL

The design and development of an application with the aforementioned characteristics involves, on one hand, segmenting both tumor and breast, and on the other, the three-dimensional reconstruction from the segmented images. A third step consists in the computation of the tumor-breast volume ratio numerical indicator using the reconstruction.

The prototype version of the system uses two mammograms in the craniocaudal and lateral views and offers the specialist the chance to identify the breast and the tumor regions by drawing, with the mouse, directly over the images. The drawing of both regions is in reality assisted by the software by means of the use of a contour detection strategy known as Intelligent Scissors [Mor95]. This technique for contour detection avoids an unwanted divergence from the targeted contour. This divergence could arise from the lack of drawing skills of the user. Once the tumor and breast are segmented in the mammography images, both images and contours are mapped into two planes in the space. These two planes can be displaced horizontally and vertically by the specialist until a location, coherent one in relation to the other, is

reached. Starting with this information, the system performs a volumetric reconstruction of the breast and tumor's bounding volume using a strategy based on contour connection [Scu89] and producing a 3D visual model of it. Additionally, the tumor-breast volume ratio is computed based on the volume reconstruction of the breast and the bounding volume of the tumor. This ratio is presented to the specialist as a numerical indicator. The system was developed using a software interactive tool implemented on the Java language and 3D graphic libraries for Java based on the standard OpenGL (Jogl) that can be run on a conventional personal computer.

#### 4. SEGMENTATION

As mentioned in the previous section, the first step in the process of reconstruction and volume estimation consists in identifying the regions corresponding to the tumor and breast in the input mammograms. After testing some segmentation schemes based on filters [Mar98] and active contours [Kas87][Dan93] with poor results, especially in the segmentation of the tumor, it was decided to implement a semi-automatic strategy allowing the specialist interaction. This strategy permitted to add expertise to the recognition process. In that sense, an interactive contour detection strategy known as Intelligent Scissors [Mor95] was adopted. This approach is considered to be in the middle between the fully automatic and the fully manual techniques that aim at contour detection. Intelligent Scissors permit the user to draw up quickly and accurately the boundaries of regions in digital images by simple movements of the mouse. When the cursor of the mouse gets close to the border of the object, the curve following the mouse movement adheres to the contour as if it had autonomous behavior (live wire). In this manner, the user does not have to trace an exact curve over the contour of the region of interest, but to drag the live wire close to the border of interest and the curve "will know" how to adhere to the correct border.

The determination of the border using Intelligent Scissors can be formulated as a problem of search in graphs where the goal is finding an optimal path between an initial and a final pixel. That is to say, the image is define as a graph  $G=(V, E)$ , where  $V$  is the set of pixels and  $E$  is the set of edges between adjacent pixels in an 8-neighborhood. Each edge has an associated cost which is computed as a local cost function involving three functional components related to the Laplacian,  $f_z(q)$ , the gradient magnitude,  $f_G(q)$ , and the gradient direction,  $f_D(p,q)$ .

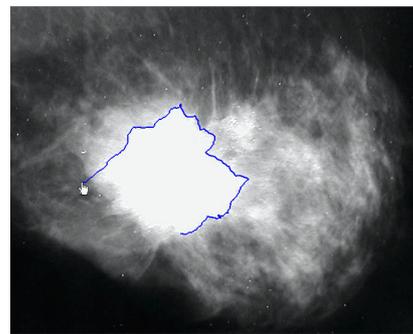
The expression for the local cost function,  $l(p,q)$  from pixel  $p$  to its neighbor  $q$  (within an 8-neighborhood), is given as follows:

$$L(p,q)=w_z \cdot f_z(q) + w_G \cdot f_G(q) + w_D \cdot f_D(p,q) \quad (1)$$

Where,  $w_z$ ,  $w_G$  and  $w_D$  are the weights associated to the corresponding characteristic function. The optimal cost is given by the minimum accumulated cost that results from adding the local costs in a path from the initial to the end pixel.

#### 5. 3D RECONSTRUCTION

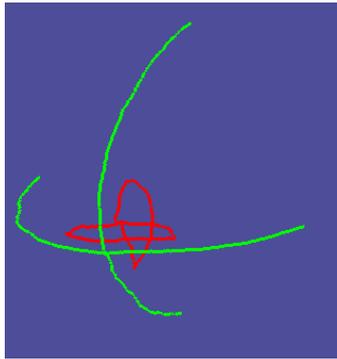
The 3D reconstruction of the tumor and breast from the mammographic views follows an ad hoc process that consists of two stages: matching and reconstruction.



**Figure 1. A "Live wire" contour is drawn by means of the intelligent scissors tool.**

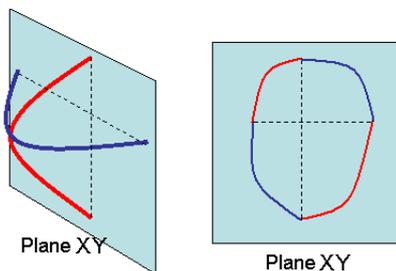
Once the tumor and breast contours are delimited in both input mammograms they are mapped into two planes that are orthogonal between them in the manner the input mammograms were acquired from the patient (craniocaudal and lateral projections). From this moment onwards, the matching stage takes place. In this stage the projected planes containing the mammograms are displaced horizontally and vertically until they reach a matching position according to the specialist judgment. Once the matching is achieved, there will be two curves in the space, associated to the contours of the mamma, and two closed contours related to the tumor. One of these curves, and its corresponding closed contour, will be placed over the XZ plane and the other over the YZ plane. An illustration of the appearance of curves and contours is shown in figure 2.

From the projected curves in spatial matching located over the XZ and YZ planes, it is possible to estimate an approximation of the original volume associated to these projections. It was decided to approximate the exterior surface of the breast assuming a semi-elliptic shape in sections of the surface parallel to its base.



**Figure 2. Result of the Matching Stage**

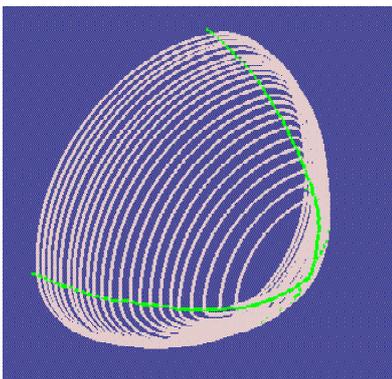
The axes of the ellipsoidal base were determined by the projections of the curves over the XY plane, such as shown in figure 3.



**Figure 3. Construction of the semi-ellipsoidal base of the mamma**

It can be noticed that the system of axes obtained by projecting the curves over the XY plane is not symmetrical therefore the semi-ellipsoidal shape is a composition of four sections of ellipses, each one of them corresponding to one quadrant.

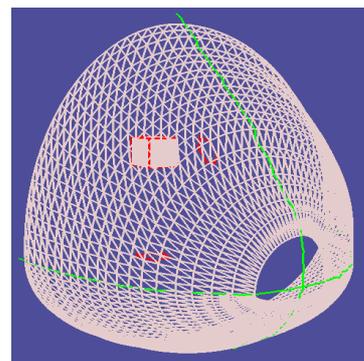
Following the outlining of the base, the projection XY plane is moved along the positive Z axis in order to get new sets of axes and semi-ellipses that approximate the peripheral surface of the mamma. The parallel semi-ellipses are shown in figure 4.



**Figure 4. 3D reconstruction of contours on the surface of the mamma.**

Once the contours are generated, a basic algorithm for contours connection [Scu89] is executed in order to produce a triangular mesh representing the approximate surface of the breast.

The representation of the tumor is approached in a different manner because tumors can occur in a multiplicity of shapes so that defining a pattern from the 2D projections is not trivial. In this work, as a first approximation, it was decided to represent the tumor by a bounding volume. With this approximation, given that the tumor is always enclosed by its bounding volume, the estimated tumor-breast volume ratio will be over the real ratio. The consequences of this coarse estimation have to be assessed before this indicator is applied in the decision making process. Different bounding volumes (box, sphere and ellipsoid) will be available for the specialist to select the most suitable for the tumor. Figure 5 shows the triangular surface of the breast obtained from the previous semi-ellipses by a connecting contour algorithm. The bounding box of the segmented tumor is also presented in the figure.



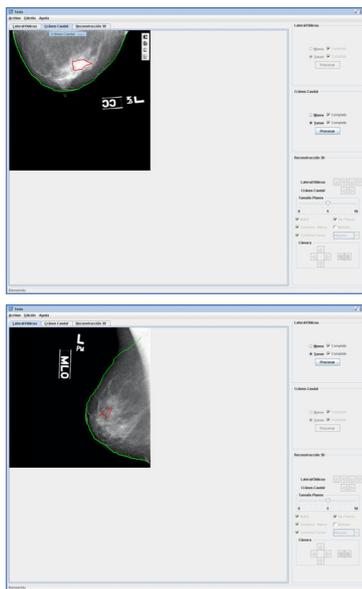
**Figure 5. 3D reconstruction of the breast's surface and bounding box of the tumor**

## 6. RESULTS

So far partial results have been obtained from the development and use of the application. It is still pending an exhaustive validation process to be carried out by the specialists. As mentioned before, the software tool was developed on Java language using Java libraries for 3D graphics based on the standard OpenGL (Jogl). The intention for this selection was to have the possibility of using it on a typical personal computer configuration. No special features in terms of graphic hardware are needed. Consistently with the process described above, the software application consists of three stages, namely, segmentation, spatial correspondence and 3D reconstruction. In the segmentation stage the breast border and the tumor closed region are interactively delimited. The spatial matching stage allows the convenient positioning of the planes. The 3D reconstruction deals with the building of a triangular

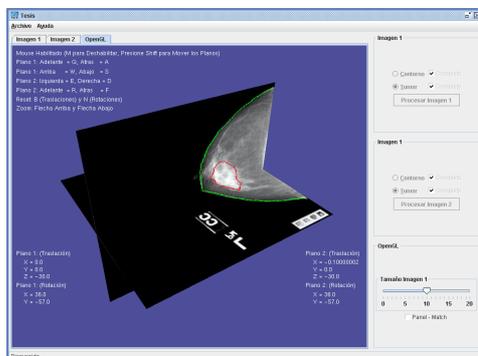
mesh from the semi-elliptical approximations of the breast surface and the determination of a bounding volume for the tumor. For each selected bounding volume a tumor-breast ratio approximation showing the numerical index on the screen.

The general appearance of the application is shown in the following figures. In summary, two areas can be distinguished, the biggest is the display area located at the left of the screen and the control area located at the right. In the display area three tags can be distinguished. The first and second tags correspond to the loading and segmentation processes for both craniocaudal and lateral mammograms (figure 6).



**Figure 6. Craniocaudal and lateral views on the display area.**

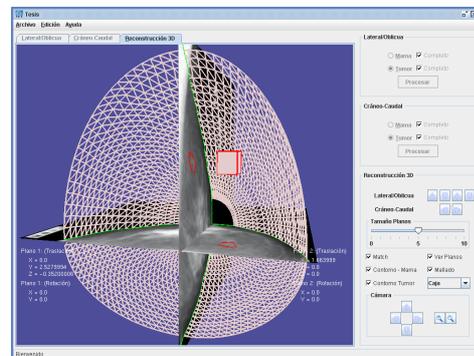
Finally, the third tag is associated to the spatial matching and 3D reconstruction of the breast and tumor (figure 7).



**Figure 7. Spatial matching and 3D reconstruction screen.**

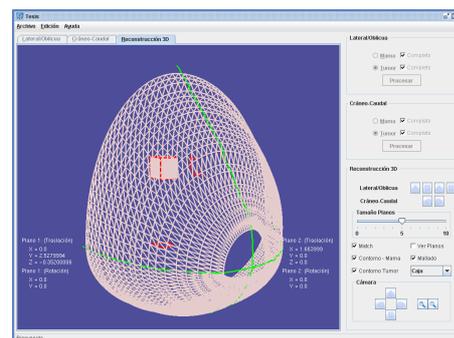
The mammograms in spatial matching, together with the contours obtained in the segmentation stage and mapped over the images, can be seen in the central display area.

A rear view of the reconstruction of a breast based on the information available in the input mammograms as well as the bounding box of the tumor is presented in figure 8.



**Figure 8. 3D reconstruction of the breast.**

Accordingly, in figure 9, a frontal view of the same reconstruction without the mammograms, so that the interest volumes are more easily distinguished, is presented.



**Figure 9. 3D reconstruction of the breast hiding the mammograms**

## 7. CONCLUSIONS

A software application intended to be used as a decision support tool for the study of the tumor-proportion from standard mammograms (craniocaudal and lateral) has been presented. The current state of development allows the specialist to delimit interactively the contours of breast and tumor within the input mammograms and establish the spatial matching between mammographic views. The software uses this segmentation and matching to carry out a 3D reconstruction of the breast and determine automatically the bounding volume of the tumor and compute a numerical indicator of the tumor-breast volume ratio between the bounding volume of the tumor and the reconstruction of the breast. This computation may prove to be of

significant value for the decision making and surgical planning. Clinical validation of the results is still pending. Accordingly, the volumetric approximation of the breast does not take into account the anatomical information available in the medio-lateral oblique mammogram which is important to incorporate the part of the breast starting in the armpit. Future developments of the application should incorporate this information.

## 8. ACKNOWLEDGMENTS

This work is part of the project “Building a visualization room”, No. 2005-12 granted by the CDCH-UC.

## 9. REFERENCES

- [Car04] Carotenuto, L., Ercolani, M., Passariello, R., et al. Software tool for computer aided diagnosis of breast cancer in contrast enhanced MR sequences. In EuroPACS-MIR 2004 in the Enlarged Europe, pages 361-364, 2004.
- [Cot05] Coto, E., Grimm, S., Bruckner, S., et al. MammoExplorer: An Advanced CAD Application for Breast DCE-MRI. In VMV'2005 Proceedings, 2005
- [Dan93] Daneels, D., Campenhout, D., Niblack, W. et al. Interactive Outlining. An improved approach using active contours, SPIE Proceedings of storage and retrieval for image and video databases, vol.1908, pp.226-233, 1993.
- [Fis02] Fisher, B., Anderson, S., Bryant, J., et al. Twenty-Year Follow-up of a Randomized Trial Comparing Total Mastectomy, Lumpectomy, and Lumpectomy plus Irradiation for the Treatment of Invasive Breast Cancer. *N Engl J Med* 2002 347: 1233-1241
- [Kas87] Kass, M., Witkin, A., Terzopoulos D.. Snakes: Active contour models, Proc. Of the First international Conference on Computer Vision, London, England, pp.259-268, 1987.
- [Kop98] Kopans, D. B., Breast Imaging, 2nd Edition, Philadelphia, Lippincott-Raven Publishers, 1998.
- [Mar80] Marr, D., Hildreth, E., A theory of edge detection, Proceedings of the Royal Society of London-Series B: Biological Sciences, vol.207, No.1167, pp 187-217, 1980.
- [Mor95] Mortensen, E., Barreto, W.. Intelligent Scissors. *Computer Graphics (SIGGRAPH '95)*, pp. 191--198 (1995).
- [Par05] Parkin, D. M., Bray, F., Ferlay, J., Pisani, P. Global Cancer Statistics 2002. *CA Cancer J Clin* 2005 55: 74-108
- [Pet03] Petroudi, S., Kadir, T., Brady, M. Automatic classification of mammographic parenchymal patterns: a statistical approach. Proc. of the 25th Annual International Conference of the IEEE, pages 798- 801, 2003.
- [Scu89] Scumaker, L. Reconstructing objects from cross-sections. *Computation of curves and surfaces*. Kluwer Academics, pages 275–309, 1989.
- [Sub04] Subramanian, K., Brockway, J., and Carruthers, W. Interactive detection and visualization of breast lesions from dynamic contrast enhanced MRI volumes. *Computerized Medical Imaging and Graphics*, 28(8):435-444, 2004.
- [Ver77] Veronesi, U., Banfi, A., Saccozzi, R., et al. Conservative treatment of breast cancer: a trial in progress at the Cancer Institute in Milan. *Cancer* 1977;39:Suppl:2822-2826.
- [Ver02] Veronesi, U., Cascinelli, N., Mariani, L., et al. Twenty-year follow-up of a randomized study comparing breast-conserving surgery with radical (Halsted) mastectomy for early breast cancer. *N Engl J Med* 2002;347:1227-1232.
- [Win92] Winchester, D. P., Cox, J. D. Standards for breast-conservation treatment. *CA Cancer J Clin* 1992 42: 134-162
- [Wir05] Wirth, M., Nikitenko, D., Lyon, J. Segmentation of the Breast Region in Mammograms using a Rule-Based Fuzzy Reasoning Algorithm. *ICGST International Journal on Graphics, Vision and Image Processing*, vol.05 (2), pages 45-54, 2005.
- [Zwi04] Zwiggelaar, R., Astley, S.M., Boggis, C. Taylor, C. Linear structures in mammographic images: detection and classification. *Medical Imaging, IEEE Transactions on Publication*. Vol. 23(9), pages: 1077-1086, 2004.

# Feature Extraction and Simplification from colour images based on Colour Image Segmentation and Skeletonization using the Quad-Edge data structure

Ojaswa Sharma  
 Department of Geodesy and  
 Geomatics Engineering,  
 University of New Brunswick,  
 Fredericton, N.-B., E3B 5A3,  
 Canada  
 ojaswa.sharma@unb.ca

Darka Mioc  
 Department of Geodesy and  
 Geomatics Engineering,  
 University of New Brunswick,  
 Fredericton, N.-B., E3B 5A3,  
 Canada  
 dmioc@unb.ca

François Anton  
 Department of Informatics and  
 Mathematical Modelling,  
 Technical University of Denmark,  
 Building 321,  
 2800 Kgs. Lyngby, Denmark  
 fa@imm.dtu.dk

## Abstract

Region features in colour images are of interest in applications such as mapping, climatology, change detection, medicine, etc. This research work is an attempt to automate the process of extracting feature boundaries from colour images. This process is an attempt to eventually replace manual digitization process by computer assisted boundary detection and conversion to a vector layer in a spatial database. In colour images, various features can be distinguished based on their colour. The features thus extracted as object border can be stored as vector maps in a spatial database after labelling and editing. Here, we present a complete methodology of the boundary extraction and skeletonization process from colour imagery using a colour image segmentation algorithm, a crust extraction algorithm and our new skeleton extraction algorithm. We also present a prototype application for completely automated or semi-automated processing of (satellite) imagery and scanned maps with an application to coastline extraction. Other applications include extraction of fields, clear cuts, clouds, as well as heating or pollution monitoring and dense forest mapping among others.

**Keywords:** colour image segmentation, polygon feature extraction, satellite imagery

## 1 INTRODUCTION

A lot of work on feature extraction from SAR (Synthetic Aperture Radar) and multi-spectral imagery has been done. A technique for coastline extraction from remotely sensed images using texture analysis is described in [3]. The delineation of the complete coastline of Antarctica using SAR imagery is shown in [15]. A morphological segmentation based automated approach for coastline extraction has been suggested in [2]. Di et al. use the image segmentation algorithm by [7] to segment an image and detect the shoreline [9].

### 1.1 Image Segmentation

The segmentation method adopted here is the one provided by [6] which is based on feature space analysis.

Feature space analysis is used extensively in image understanding tasks. Comaniciu and Meer [6] provide a comparatively new and efficient segmentation algorithm that is based on feature space analysis and relies on the *mean-shift algorithm* to robustly determine the

cluster means. A *feature space* is a space of feature vectors. These features can be object descriptors or patterns in the case of an image. As an example, if we consider a colour image having three bands (red, green, and blue), then the image we see as intensity values plotted in Euclidean XY space is said to be in *image space*. Consider a three dimensional space with the axes being the three bands of the image. Each colour vector corresponding to a pixel from the image can be represented as a point in the feature space.

Given  $n$  data points  $\vec{x}_i$ ,  $i = 1, \dots, n$  in the  $d$ -dimensional space  $\mathbb{R}^d$ , a *flat kernel*, that is a characteristic function of the  $\lambda$ -ball in  $\mathbb{R}^d$ , is defined as:

$$K(\vec{x}) = \begin{cases} 1 & \text{if } \|\vec{x}\| \leq \lambda \\ 0 & \text{if } \|\vec{x}\| > \lambda \end{cases} .$$

The *mean shift* vector at a location  $\vec{x}$  is defined as:

$$M_\lambda(\vec{x}) = \frac{\sum_{\vec{r} \in \mathbb{R}^d} K(r - \vec{x}) \vec{r}}{\sum_{\vec{r} \in \mathbb{R}^d} K(r - \vec{x})} - \vec{x} .$$

In his work, Cheng shows that the mean shift vector, the vector of difference between the local mean and the center of the window  $K(\vec{x})$ , is proportional to the gradient of the probability density at  $\vec{x}$  [5]. Thus mean shift is the steepest ascent with a varying step size that is the magnitude of the gradient. The fundamental use

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Copyright UNION Agency - Science Press, Plzen, Czech Republic.

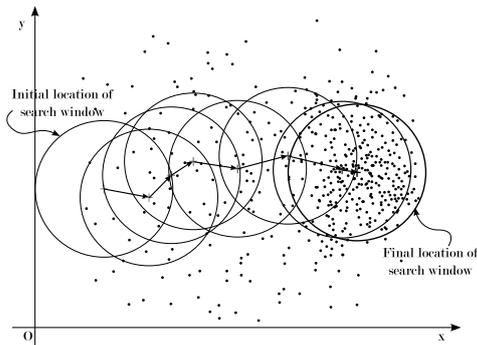


Figure 1: Mode seeking using mean shift algorithm.

of mean shift is in seeking the modes (i.e. the most frequent value occurring in a sampling of a random variable) that give regions of high density in any data. Comaniciu and Meer use mean shift vector in seeking the mode of a density by shifting the kernel window by the magnitude of the mean shift vector repeatedly [7]. The authors also prove that the mean shift vector converges to zero and eventually reaches the basin of attraction of that mode. This is shown graphically in Figure 1 where the initial position of the kernel window is chosen at a location far from the mode of the data and the window is moved by a magnitude equal to that of the mean shift vector at the current window location in each step. It can be seen that the mean shift vectors gradually converge as the window moves near the maximum density region.

In their research work, Comaniciu and Meer state a simple, adaptive steepest ascent mode seeking algorithm [6].

1. Choose the radius  $r$  of the search window (i.e. radius of the kernel).
2. Choose the initial location of the window.
3. Compute the mean shift vector and translate the search window by that amount.
4. Repeat until convergence.

The mean shift algorithm gives a general technique of clustering multi-dimensional data and is applied here in colour image segmentation.

The method described in [6] provides an autonomous segmentation technique with only the type of segmentation to be specified by the user. This method emphasizes the importance of utilizing the image space along with the feature space to efficiently perform the task of segmentation. The segmentation has three characteristic input parameters:

- Radius of the search window,  $r$ ,
- Smallest number of elements required for a significant colour,  $N_{min}$ , and

- Smallest number of connected pixels necessary for a significant image region,  $N_{con}$ .

The size of the search window determines the resolution of the segmentation, smaller values corresponding to higher resolutions. The authors use square root of the trace of global covariance matrix of the image,  $\sigma$ , as a measure of the visual activity in the image. The radius  $r$  is taken proportional to  $\sigma$ . For the implementation of the segmentation algorithm, the authors provide three segmentation resolution classes:

1. **Undersegmentation** refers to the lowest resolution with a minimum number of colours and only dominant regions of the image. The three parameters for this class are:  $(0.4\sigma, 400, 10)$ .
2. **Oversegmentation** refers to intermediate resolution and represents objects with some level of detail. The three parameters for this class are:  $(0.3\sigma, 100, 10)$ .
3. **Quantization** refers to the highest resolution and produces image with all the important colours with no object connectivity requirement. The three parameters for this class are:  $(0.2\sigma, 50, 0)$ .

Figure 2 shows the results of this segmentation algorithm on a natural image. Note the variation in number of colours for each segmentation type.

Later, Comaniciu and Meer provide an improvement [7] over this segmentation algorithm by merging the image domain and the feature (range) space into a joint spatial-range domain of dimension  $d = p + 2$ , where  $p$  is the dimension of the range domain. This gives an added advantage of considering both spaces together and gives good results in cases where non-uniform illumination produces false contours when the previous segmentation algorithm is used. Therefore, the new algorithm is particularly useful to segment natural images with man-made objects. An added computational overhead to process higher dimensional space is inevitable here. In this research, since we are dealing with scanned maps, the simple mean shift based segmentation algorithm provides satisfactory results.

## 1.2 Boundary Extraction

The work presented in [1] leads to the extraction of object boundary from a set of sufficiently well sampled data points. The vertices of the Voronoi diagram approximate the medial axis of a set of sample points from a smooth curve. Vertices of the Voronoi diagram of the sample points were inserted into the original set of sample points and a new Delaunay triangulation was computed [1]. The circumcircles of this new triangulation approximate empty circles between the original boundary of the object and its skeleton. Thus, any Delaunay



(a) Original image with 108440 colours



(b) Undersegmented image with 8 colours



(c) Oversegmented image with 34 colours

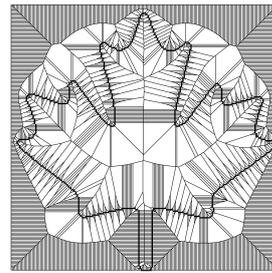


(d) Quantized image with 49 colours

Figure 2: Colour image segmentation by [6]

edge connecting a pair of the original sample points in the new triangulation is a part of the border [1].

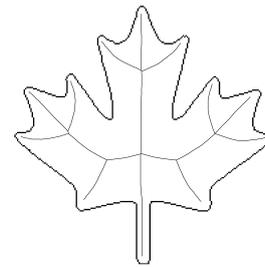
Further research by [10] leads to a One-step border (crust) extraction algorithm. In a Delaunay triangulation, each Delaunay edge is adjacent to two triangles and the circumcenters of these triangles are the Voronoi vertices. A Voronoi edge connecting these two circumcenters is the dual edge to the Delaunay edge considered here. According to [10], a Delaunay edge is a part of the border if it has a circle that does not contain any Voronoi vertex. It is sufficient to test only the vertices of the dual Voronoi edge. The test is the standard *InCircle* test. Considering two



(a) The Voronoi diagram



(b) Anti-crust



(c) Skeleton

Figure 3: Skeleton as seen as the anti-crust.

triangles  $(p, q, r)$  and  $(r, q, s)$  sharing an edge  $(q, r)$  in a Delaunay triangulation and letting  $v$  be a vector orthogonal to edge  $(r - q)$  in clockwise order, the test is:  $(\vec{q}\vec{s} \cdot \vec{r}\vec{s})(\vec{q}\vec{p} \cdot \vec{r}\vec{p}) \geq -(\vec{r}\vec{s} \cdot v)(\vec{q}\vec{p} \cdot v)$ . This test will be true for an edge in the border set. Furthermore, those Delaunay edges that are not part of the border set have their dual Voronoi edges as being part of the skeleton.

### 1.3 Skeleton Extraction

Popular methods of skeleton extraction are thinning using mathematical morphology [13, chap. 9] and skeletonization using distance transform [4]. This research is concerned with skeleton extraction using the Voronoi diagram [1, 10, 16, 18].

The work presented in [1] shows that the “crust” or the boundary of a polygon can be extracted from an unstructured set of points provided the data points are well sampled. Gold and Snoeyink [11] further simplify their method and show that the boundary can be extracted in a single step (see section 1.2). Gold [10] discusses about “anti-crust” in context of skeleton extraction citing a brief introduction of this term in [1]. The idea behind getting skeleton is that a Voronoi edge is a part of the skeleton if its corresponding dual Delaunay edge is not a part of the border set (crust) and it lies completely within the selected object. Thus, selecting the Voronoi edges lying inside the selected object that are dual of the non-crust Delaunay edges should give us the skeleton (see Figure 3). The Voronoi edges thus selected form a tree structure called the “anti-crust” [10] that extend towards the boundary but do not cross it.

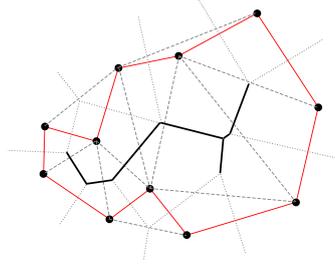


Figure 4: Anti-crust from the crust.

The anti-crust of an object, as described above, forms a tree like structure that contains the skeleton. Once all the Delaunay edges belonging to the border set or the crust are identified using the condition given by [10], it is easy to identify the Voronoi edges belonging to the anti-crust. In Figure 4, consider the Delaunay triangulation (dashed edges), the corresponding Voronoi diagram (dotted edges) and the crust edges (solid red edges).

Navigation from a Delaunay edge to its dual Voronoi edge can be achieved by using the  $Rot()$  operator in the quad-edge data structure. A Voronoi edge  $e.Rot()$  of the dual Delaunay edge  $e$  is marked as an edge belonging to the anti-crust if the following conditions are satisfied:

1.  $e \notin Crust$
2.  $e.Rot().Org \in I$
3.  $e.Rot().Dst \in I$

Where  $e.Rot().Org$  is the origin coordinate of edge  $e.Rot()$ ,  $e.Rot().Dst$  is the destination coordinate of edge  $e.Rot()$  and  $I$  is the selected object. This marks all the Voronoi edges belonging to the anti-crust that fall inside the selected object. Negating conditions (2) and (3) so that the coordinates do not fall inside the object will give us the exterior skeleton or the *exoskeleton*. Once the anti-crust is identified, an appropriate pruning method can be applied to get rid of the unwanted edges.

## 1.4 Skeleton Pruning

The “hairs” around the skeleton result from the presence of three adjacent sample points whose circumference does not contain any other sample point - either near the end of a main skeleton branch or at locations on the boundary where there is minor perturbation because of raster sampling [10]. A skeleton retraction scheme suggested by [12] gets rid of the hairs and also results in smoothing of the boundary of the object. Ogniewicz [17] presents an elaborate skeleton pruning scheme based on various residual functions. Thus a hierarchic skeleton is created which is good for multiscale representation. The problem of identifying skeleton edges now reduces to reasonably prune the anti-crust.

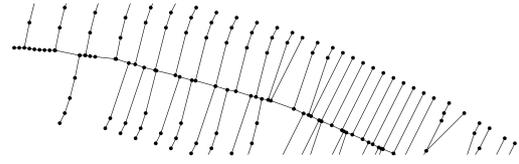


Figure 5: Hair around the skeleton composed of multiple edges.

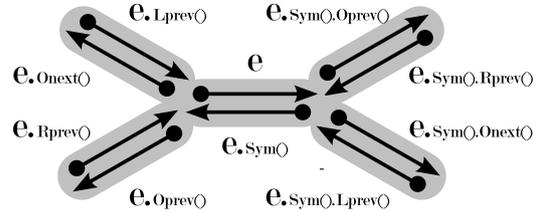


Figure 6: Accessing neighboring edges in a quad-edge.

Gold and Thibault [12] present a retraction scheme for the leaf nodes in the anti-crust. The skeleton is simplified by retracting the leaf nodes of the skeleton to their parent nodes. Gold and Thibault [12] recommend performing the retraction operation repeatedly until no further changes take place. An observation reveals that an unwanted branch in a skeleton may be composed of more than one edge (see Figure 5). Therefore, single retraction is not sufficient to provide an acceptable skeleton.

## 2 OUR SKELETON APPROACHES

This is our main contribution together with the integration of this skeleton approach with existing techniques described above. We address the problem mentioned in Section 1.4 by pruning the leaf edges instead of retracting the leaf nodes. Leaf edge pruning produces satisfactory results and requires only two or three levels of pruning. Before pruning the leaf edges, these must be identified in the anti-crust using the operations provided by the quad-edge data structure (see Figure 6).

An edge  $e$  from a tree of edges  $T \in V$ , where  $V$  is the Voronoi diagram, is marked as a leaf edge if the following condition is satisfied:

$$e.Oprev() \notin T \text{ And } e.Onext() \notin T$$

Or

$$e.Sym().Oprev() \notin T \text{ And } e.Sym().Onext() \notin T.$$

This condition essentially selects all the Voronoi edges belonging to the anti-crust that have at least one end point free (i.e., connected to an edge not belonging to the anti-crust). This condition is used to locate leaf edges followed by their removal from the skeleton. Experiments show that removing leaf edges two to three times simplifies the skeleton to a major extent for linear features.

Finally, we have found a labelling scheme for the vertices of the border set that allows us to get the skeleton

without hairs and gaps from the inner Delaunay triangulation. We identify the vertices of the border set that are incident to two Delaunay edges of the border set. We call them branching vertices. We start to give numerical labels to the vertices of the border set starting at a branching vertex, and we increment the label each time we encounter a new branching vertex. On the example of Figure 4, there are three branching vertices. We get three labels 1, 2 and 3, and three branches for the skeleton. We keep only Voronoi edges between vertices having different labels.

### 3 AUTOMATED APPROACH TO SKELETONIZATION

The general approach adopted here is:

1. Segment a colour image into prominent objects.
2. Ask the user if all the objects are to be processed independently (automatic process) or an individual object (semi-automatic process) selected.
3. Collect sample points for each object to be processed.
4. Extract the skeletons using Delaunay/Voronoi diagram based algorithm.

Once objects are defined as homogeneous regions by the segmenter, the next step is to either select them all or some of them. To achieve this, the user is allowed to select a region on the image. If an object is composed of more than one regions then multiple object selection can be made and regions combined to form a single object. A wrongly selected region can be removed from the selection. The user input is processed and the selected region is highlighted and selected for next processing.

Once we have an object or all the objects chosen from an image, the next step is to sample its boundary in order to generate points used to construct the Delaunay triangulation. In order to automatically generate these sample points, edge pixels that are returned by a morphological edge detector are used. Using edge pixels also helps in generating a dense sampling which is required to give a better approximation of the skeleton [1]. Morphological edge detection on the binary image containing the selected object is performed and the edge pixels are then sequentially inserted into the Delaunay triangulation. The triangulation is updated after every insertion (using the incremental algorithm).

The Delaunay triangulation of the sample points is computed using the incremental algorithm given by [14] which is stored in the quad-edge data structure. This is followed by computation of the Voronoi vertices for all faces of the triangulation. The boundary of the object is extracted using the criterion given by [10]. The edges in the Delaunay triangulation are analyzed and flagged as being part of the boundary.

## 4 RESULTS WITH SATELLITE IMAGERY

In the following example, the coastline is extracted as the boundary of the selected object. The accuracy of the coastline rendition depends on the spatial resolution of the imagery.

Since the system is designed to make use of homogeneity in colors of objects, natural objects are better suited for our analysis. A few cases of coastline extraction have been considered here. Figure 7 shows the complex coastline of Guinea Bissau. Segmentation results in four colours that define the water body out of which three define the coastline. Multiple selection enables combining these three regions together to form the complete coastline as shown in Figure 7(d).

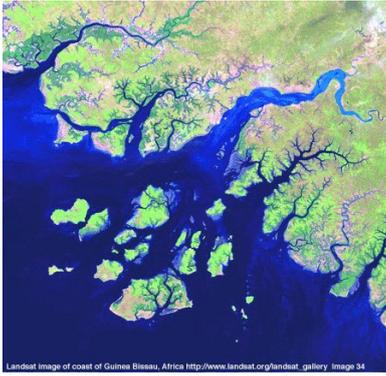
Satellite image of Lake Jempang, East Kalimantan, Indonesia (20 Jul 2002) obtained on-line from Centre for Remote Imaging, Sensing and Processing ([http://www.crisp.nus.edu.sg/monthly\\_scenes/y2002/Jul02\\_i.html](http://www.crisp.nus.edu.sg/monthly_scenes/y2002/Jul02_i.html)) is shown in Figure 8(a). The extracted lake boundary is shown in Figure 8(d).

The map in Figure 9(a) is obtained processing the photograph from the proposal for Decision Support for Flood Event Prediction and Monitoring (FEPM) project set forth by the Emergency Measures Organization, New Brunswick, Canada. It shows flood risk mapping from the 1973 flood in the Saint John river in Fredericton, New Brunswick and the risk of flooding in the area in a time span of 20 years (extracted in Figure 9(c) and (d)), 100 years (extracted in Figure 9(e) and (f)) and actual flood extent occurred in the year 1973 (extracted in Figure 9(g) and (h)). The flood plains are extracted as polygon vector boundaries from crust of the selected object.

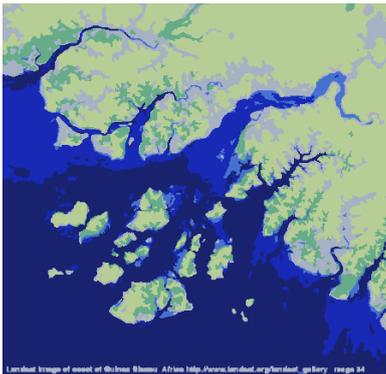
Figure 10(a) shows small portion of a map of the city of Moncton, New Brunswick, Canada (Map Number - 211/02) scanned from the *New Brunswick Atlas* (second edition). The map is scanned at 300 DPI (dots per inch) resolution and 24-bit colour depth. The map mainly consists of three types of roads: national highway (green), highway (blue) and streets (red). Crisp road edges with well defined road patches can be seen in the segmented image (Figure 10(b)) that are extracted as skeletons (see Figures 10(d) and (f)). The skeletons show the presence of minor clutter due to misclassification during the segmentation process. Such a problem can be easily rectified in a GIS environment while manually editing the data after import.

## 5 TIME COMPLEXITY

It is important to analyse the complete procedure for time complexity. Since the overall complexity will depend on the intermediate steps, these are first analysed individually. Comaniciu [8, p. 21] shows that the complexity of the probabilistic mean shift type algorithm



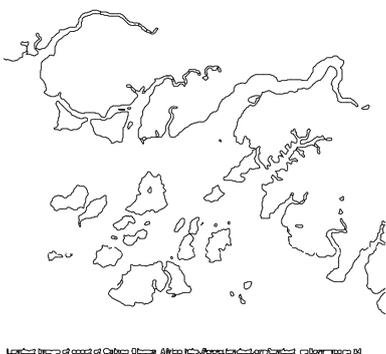
(a) Satellite image of Guinea Bissau



(b) Over-segmented image



(c) North Atlantic ocean

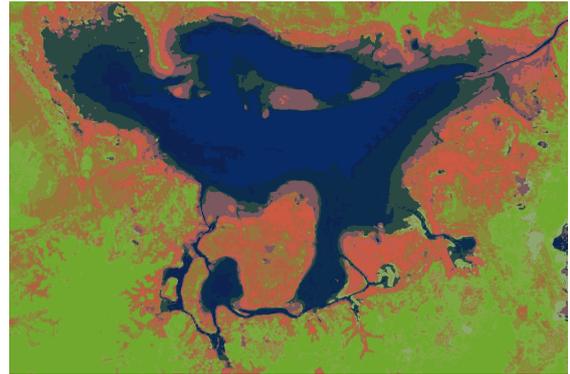


(d) Extracted coastline

Figure 7: Feature boundary extraction from the satellite image of Guinea Bissau.



(a) Lake Jempang



(b) Segmented image



(c) Selected lake



(d) Lake boundary

Figure 8: SPOT satellite image of Lake Jempang, East Kalimantan, Indonesia (20 Jul 2002).

that is employed in the segmentation algorithm [6] is  $O(mn)$ , with  $m \ll n$  where  $n$  is the number of pixels in the input image (or the number of feature vectors in the feature space) and  $m$  is the number of vectors in initial feature palette or clusters. Comaniciu [8, p. 29] claims that the segmentation algorithm is linear with the number of pixels in the image.

Object selection is implemented as a simple search of pixel colour values compared with the user selected points on the image. Therefore, it is  $O(n)$  with the number of pixels  $n$  in the image.

Binary edge detection is implemented as subtraction of two binary images. One of the binary images is always the original image while the other is either a dilated or an eroded image of the original. For an image  $I$  of size  $x \times y$  and a structural element of size  $k \times l$ , computation of either erosion or dilation requires  $x \times y$  iterations, each requiring  $(k \times l) + 1$  comparisons. Therefore, the time complexity is linear with number of pixels of the input binary image and linear with the area of the structural element. Overall time complexity can be said to be  $O(mn)$  where  $n$  is number of pixels in the image and  $m$  is number of pixels in the structuring element. Further, in the structural element, only locations having a value of 1 are considered in the computations. In our case, the application uses a  $3 \times 3$  structural element with only five values being 1's in the mask (i.e.,  $m = 5$ ). Therefore, we can safely say that edge detection is  $O(n)$  with the number of pixels  $n$  in the image.

## 6 CONCLUSIONS

This research work succeeds in achieving its primary goals by designing an effective methodology for automated vectorization of features in colour images. The methodology enables extraction of boundaries of an object in a single step.

Based on the methodology, an interactive software application has been developed. It incorporates object extraction from input images using colour image segmentation. Segmentation based on clustering using a mean shift algorithm in feature space has been adopted here. The mean shift algorithm is a popular and robust method of clustering and provides good results in segmentation. The application allows selection of multiple objects for extraction of the boundary.

The applicability of the methodology to colour images has been shown by extracting natural or artificial features from (satellite) images. Applicability of the developed methodology can be easily extended to natural colour satellite imagery to extract homogeneous features. Coastline delineation, snow cover mapping, cloud detection, and dense forest mapping are a few areas where satisfactory results can be obtained.

## REFERENCES

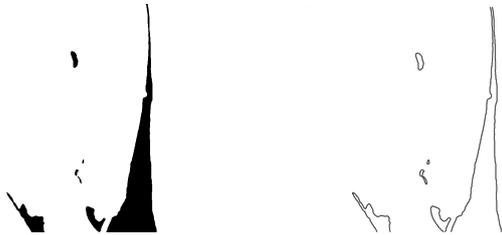
- [1] N. Amenta, M. Bern, and D. Eppstein. The crust and the  $\beta$ -skeleton: Combinatorial curve reconstruction. *Graphical models and image processing: GMIP*, 60(2):125–135, 1998.
- [2] S. Bagli and P. Soille. Morphological automatic extraction of coastline from pan-european landsat tm images. In *Proceedings of the Fifth International Symposium on GIS and Computer Cartography for Coastal Zone Management*, volume 3, pages 58–59, Genova, 2003.
- [3] G. Bo, S. Delleplane, and R. De Laurentiis. Coastline extraction in remotely sensed images by means of texture features analysis. In *Geoscience and Remote Sensing Symposium, IGARSS '01*, volume 3, pages 1493–1495, Sydney, NSW, Australia, 2001.
- [4] G. Borgefors. Distance transformations in arbitrary dimensions. *Computer Vision, Graphics, and Image Processing*, 27(3):321–345, 1984.
- [5] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995.
- [6] D. Comaniciu and P. Meer. Robust analysis of feature spaces: color image segmentation. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, pages 750–755, Washington, DC, USA, 1997. IEEE Computer Society.
- [7] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 24(5):603–619, 2002.
- [8] Dorin I. Comaniciu. *Non-Parametric Robust Methods for Computer Vision*. PhD thesis, Rutgers, The State University of New Jersey, 2000.
- [9] K. Di, J. Wang, R. Ma, and R. Li. Automatic shoreline extraction from high-resolution ikonos satellite imagery. In *Proceeding of ASPRS 2003 Annual Conference*, volume 3, Anchorage, Alaska, 2003.
- [10] Christopher M. Gold. Crust and anti-crust: A one-step boundary and skeleton extraction algorithm. In *Symposium on Computational Geometry*, pages 189–196, New York, NY, USA, 1999. ACM Press.
- [11] Christopher M. Gold and Jack Snoeyink. A one-step crust and skeleton extraction algorithm. *Algorithmica*, 30(2):144–163, Jun 2001.
- [12] Christopher M. Gold and David Thibault. Map generalization by skeleton retraction. In *Proceedings of the 20th International Cartographic Conference (ICC)*, pages 2072–2081, Beijing, China, August 2001.
- [13] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Prentice Hall, 2 edition, 2002.
- [14] P.J. Green and R. Sibson. Computing dirichlet tessellations in the plane. *The Computer Journal*, 21(2):168–173, 1977.
- [15] H. Liu and K. C. Jezek. A complete high-resolution coastline of antarctica extracted from orthorectified radarsat sar imagery. *Photogrammetric Engineering and Remote Sensing*, 70(5):605–616, 2004.
- [16] R. Ogniewicz. Automatic medial axis pruning by mapping characteristics of boundaries evolving under the euclidean geometric heat flow onto Voronoi skeletons. Technical Report 95-4, Harvard Robotics Laboratory, 1995.
- [17] R. L. Ogniewicz. Skeleton-space: A multiscale shape description combining region and boundary information. In *Proceedings of Computer Vision and Pattern Recognition, 1994*, pages 746–751, 1994.
- [18] R. L. Ogniewicz and O. Kübler. Hierarchic Voronoi skeletons. *Pattern Recognition*, 28(3):343–359, 1995.



(a) Flood extents in the city of Fredericton



(b) Over segmented image 11 colours



(c) 20 years flood zone

(d) extracted boundary

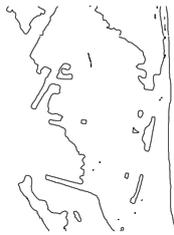


(e) 100 years flood zone



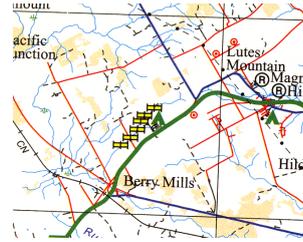
(f) extracted boundary

(g) 1973 flooded area

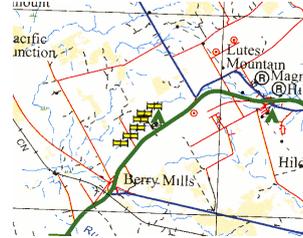


(h) extracted boundary

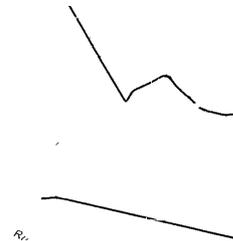
Figure 9: A portion of photographed flood risk map showing flood extents.



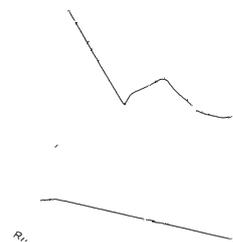
(a) Map with 62800 colours



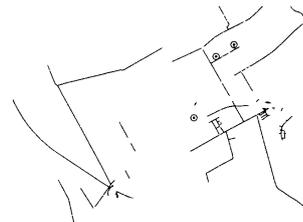
(b) Segmented image with 10 colours



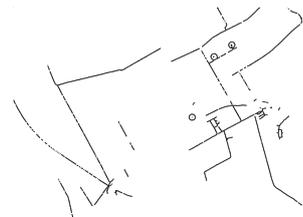
(c) Highway



(d) Skeleton of highway



(e) Streets



(f) Skeleton of streets

Figure 10: Extraction of various roads from a map.

# Associating 6 DoF sensor data to 3D scan view registration

Joris Vergeest  
Delft University of Technology  
Landbergstraat 15  
2628CE Delft, The Netherlands  
J.s.m.vergeest@tudelft.nl

Thomas Kroes  
Delft University of Technology  
Landbergstraat 15  
2628CE Delft, The Netherlands  
thomaskroes@wanadoo.nl

Wolf Song  
Delft University of Technology  
Landbergstraat 15  
2628CE Delft, The Netherlands  
y.song@tudelft.nl

## ABSTRACT

To make 3D scanning an attractive tool for incidental or inexperienced users, the process of scan view registration should be avoided or significantly simplified. If a 6 DoF sensor is attached to the object to be scanned, additional data about each of the 3D views can be supplied to the registration software as to provide initial relative placements of pair of views, thus making automatic matching feasible. This releases the user from the tedious manual registration process. A method to apply a 6 DoF device to 3D scanning is in development. To calibrate the sensor to the scanner, the equation of similarity matrices needs to be solved. We verified numerically that this leads to ambiguities if only one sensor-to-scanner association is measured. A method based on a geometric treatment is proposed to achieve an unambiguous association between the two devices. Initial numerical results are presented.

## Keywords

Conceptual shape design, 3D scanning, calibration, registration, 6 DoF sensor

## 1. INTRODUCTION

In industrial design engineering 3D scanning is increasingly used to provide a starting point in conceptual shape design. Rather than creating a new product's shape from scratch with a CAD system, designers may chose to first create a physical model manually (e.g. made out of clay), obtain a surface mesh from the model using 3D scanning and then import the surface mesh into a CAD modeling system for further modification and refinement. Also the redesign of manufactured parts or the reuse of existing product shape features is becoming routinely applied in product design [Smyth 2000, Vergeest 2001, Song 2005].

To generate a CAD surface or solid model from a physical object, the designer (or user, in general) needs to take the following steps. First the object's surface is digitized using some digitization instrument. We use the Minolta Vivid 700 scanner, which produces almost instantly, a matrix of (at most)

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.  
Copyright UNION Agency – Science Press, Plzen, Czech Republic.

200×200 three-dimensional data points of the part of the surface which is orientated toward and visible by the scanner, from a particular viewing direction. Multiple views should be taken from other directions, until the entire surface has been recorded. In case consumer products are digitized, which have a diameter typically in the range between 5cm to 50cm, the user should displace and/or rotate the product between subsequent shots. In most cases, taking the scanning views can be routinely performed by people without experience of using the scanner.

The different data sets, or point clouds, one for each scan view, need to be assembled into one set, in a common coordinate system. This process is called scan view registration. Normally, the user selects two views which are partly overlapping. Graphically supported by a software tool belonging to the scanning system, the user must approximately designate corresponding points in each of the two views, in the overlap region. With three or more of such point pairs, the software derives an initial relative positioning of the two views, and an algorithm will then search for the pose of maximum matching of the point sets in their overlap region. If the algorithm fails to find the optimal pose, the user is prompted to designate new or additional point pairs. If the algorithm succeeds in finding a good match, the next scan view is considered, which should be

matched with the views already processed. When all scan views are thus processed, they are defined (at least approximately) in the same coordinate system. Finally, the scanner's software performs a global registration, in which all scan views are considered collectively in order to obtain the best achievable placements of the scan views relative to each other.

The resulting point set may then be converted into a surface mesh, from which a CAD surface model can be derived, or in case the mesh is consistent with a topological shell structure, a CAD solid model. From then on, the model is in a format to which designers are accustomed.

The major bottleneck in this process is the scan view registration. For an inexperienced user the designation of corresponding points in different views is difficult and the results of the registration software is often unpredictable. As a result, 3D scanning is perceived as unpractical to (*e.g.*) shape designers. This problem disappears when the scanned object is stationary and the scanning device's position and orientation is tracked, for example using a mechanical arm or another tracking system. Then the registration procedure can be practically automated based on the tracking data. However, this method is designed for hand-held scanners and large objects, like cars or statues, and the devices are relatively costly. To scan small objects, a rotation platform can be used to bring the object in different orientations. Since the rotation axis is mechanically very stable, and sometimes the amount of rotation is known to the matching software as well, automatic registration can be achieved. However, the use of a rotation table limits the orientation of the object relative to the scanner, which is unwanted when the entire surface needs to be digitized.

We experienced that designers find it quite natural to hold an object (for example a clay model they produced manually) in front of the scanner in different orientations, thus collecting scan views from the entire surface. However, as mentioned, the next step, which is registration of the scan views into a single surface mesh, is perceived as too complex and too tedious.

We have developed a solution to this problem, based on additional data from a 6 degrees-of-freedom (DoF) positioning sensor attached to the object being digitized. To correctly associate the 6D placement data from the sensor with the required matching transformation of the 3D scan views, the sensor should be calibrated to the scanning device. We can derive the calibration from a couple of scan views, such chosen that they can be matched easily. The matching transformations and their corresponding 6D displacements (from the sensor) form, pair wise, so-

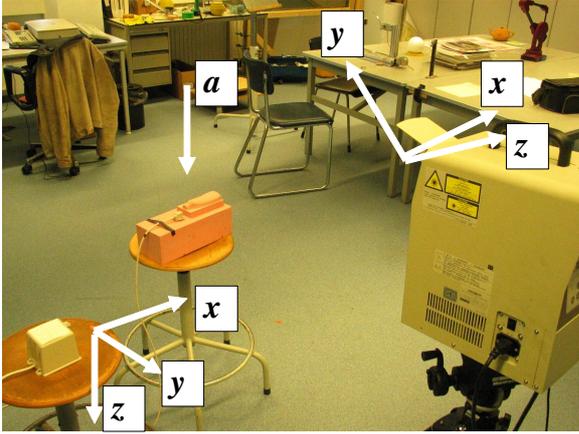
called similar matrices. The calibration transform is then equal to the quotient from any such pairs. This type of calibration is different from calibrations based on position/orientation information from each of the two reference frames, as is the case, for example, in calibration problems for Augmented Reality devices [Grasset 2001], [Kato 1999], [Wheeler 1998]. Our problem would be similar to those when we would base the scanning information on explicit feature points on or near the sensor. In our proposed method however, we neither need to rely on feature recognition nor to make assumptions about the sensor's centre relative to its housing. We only need to perform some relatively simple scan view matchings.

In this paper we present in Section 2 the mathematical adaptation of the sensor data to the registration algorithm as a minimization problem and we analyze the problem using eigenvectors of the transforms. In section 3 we verify numerically how the displacement of the 6 DoF sensor, should be associated to the scan registration matrix. The derivation of the calibration matrix is presented in Section 4. In Section 5 we sketch a method to apply the 6 DoF sensor data to the registration algorithm. Conclusions are drawn in Section 6.

## 2. ASSOCIATING THE 6D DATA TO THE REGISTRATION TRANSFORM

The sensor is a device that measures its own position and orientation relative to a magnetic transmitter, which is stationary in the laboratory. In our application, also the 3D scanning device is stationary relative to the laboratory. Therefore, if the sensor is rigidly connected to the object being digitized (see Fig. 1), it should be possible to determine the placement of the object relative to the scanner as a function of time.

The sensor produces 6-tuples  $(x(t), y(t), z(t), \alpha(t), \beta(t), \gamma(t))$ , its position coordinates and orientation angles, as a function of time, relative to the magnetic transmitter. Obviously a calibration step is required to define a "starting" placement of the sensor at time  $t_0$  relative to the scanner. All placements at  $t > t_0$  are then defined relative to the starting placement. We need to determine the coordinate system  $T$  of the magnetic transmitter relative to the coordinate system  $S$  of the scanner, *i.e.* we need to find  ${}^S T$  (the upper index denoting the reference frame).



**Figure 1. Calibration setup: scanner and its local frame  $S$  on the right-hand side, transmitter of 6 DoF sensor and its frame  $T$  on the left and scanned object in the center. The 6 DoF sensor (wire coming out) is attached to the object during (at least) two subsequent scan views taken.**

A 6-tuple delivered by the magnetic sensor defines, by convention, an equivalent  $4 \times 4$  matrix  ${}^T F$  which can be interpreted as the local frame of the sensor measured relative to frame  $T$ . If two frames  ${}^T F(t_0)$  and  ${}^T F(t_1)$  are measured,  $t_1 > t_0$ , then the 6-dimensional displacement of the sensor over the time interval is  ${}^T F(t_1) ({}^T F(t_0))^{-1}$ . Equivalently, the transformation to bring  ${}^T F(t_1)$  back to  ${}^T F(t_0)$  is defined as  $N = {}^T F(t_0) ({}^T F(t_1))^{-1}$ . Now we need to relate these measurements to measurements relative to the scanner. A problem is that the scanner does not deliver quantities like  ${}^S F(t_0)$ . All we get from the scanner is a set of 3D data points defined relative to  $S$ , but there is no explicit information about  $S$ . However, if we apply the registration procedure of the scanner to two point sets, one from the object (including the sensor) at  $t_0$  and one at  $t_1$ , then the outcome of the registration is the displacement  $M$ , where  $M = {}^S F(t_0) ({}^S F(t_1))^{-1}$ . If transformation  $M$  is applied to the points obtained at  $t_1$  then the transformed points will be in accordance with the points obtained at time  $t_0$ . The matrices  $N$  and  $M$  describe the same displacement, defined relative to  $T$  and to  $S$ , respectively. Such matrices are called *similar matrices*, for which exists a similarity transformation  $X$  such that  $XNX^{-1} = M$ .  $X$  specifies coordinate system  $T$  relative to  $S$ , or  $X = {}^S T$ , exactly the quantity we were looking for. This can be verified as follows:

$$\begin{aligned} XNX^{-1} &= {}^S T {}^T F(t_0) {}^{F(t_1)} T {}^T S \\ &= {}^S F(t_0) {}^{F(t_1)} S = {}^S F(t_0) ({}^S F(t_1))^{-1} = M. \end{aligned} \quad (1)$$

The calibration comes down to finding  $X$  for given  $M$  and  $N$ . If the sensor placements could be measured

with infinite precision and if the scanning registration would be perfect, equation (1) could be solved for  $X$  based on a single observation of  $M$  and  $N$  (however, we will show that the solution is not unique). In practice we measure a set of  $n$  pairs  $(M_i, N_i)$  and search for the 6-tuple  $X' = (x_x, y_x, z, \alpha_x, \beta_x, \gamma_x)$  which minimizes

$$d = \sum_{i=1}^n |XN_i X^{-1} M_i^{-1}|^2, \quad (2)$$

where  $X$  denotes the placement matrix derived from  $X'$ . The norm in equation (2) can be defined as a function of the principle rotation angle and displacement component of the  $4 \times 4$  matrix. In the next section we will numerically verify that equation (2) has no unique solution for  $n=1$ , and we will present a solution to it for small  $n, n > 1$ .

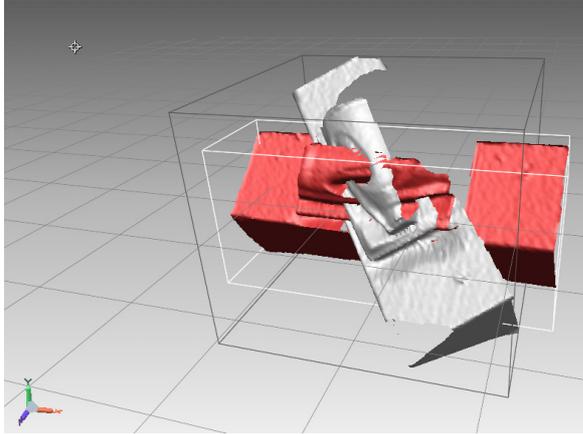
### 3. NUMERICAL QUANTITIES

We consider two placements of the object shown in Fig 1. The 3D scanner (in this setup a Minolta Vivid 700 [Min2006]) and the 6 DoF sensor (Flock of Birds, or FOB, from Ascension [Ase2006]) attached to the object are shown. The placement matrix of the FOB sensor is measured relative to the magnetic transmitter, shown in Fig.1 together with an indication of its coordinate system. The difference between the two placements of the objects was chosen to be roughly an anti-clockwise rotation (as viewed in Fig.1) of the object about a vertical axis near the center of the object. The approximate axis direction is indicated by vector  $a$  in the picture. The placement difference can be viewed in Fig. 2, which shows the original scan data views  $V_1$  and  $V_2$ . The RapidForm software [Rap 2006] was used to create the pictures and to perform scan view registration. The registration was implemented as a transformation  $M$  to displace  $V_2$  back to  $V_1$ , hence, roughly, a clockwise rotation is applied to  $V_2$ , which is equivalent by rotation in positive direction about axis  $a$ , directed vertically downward.

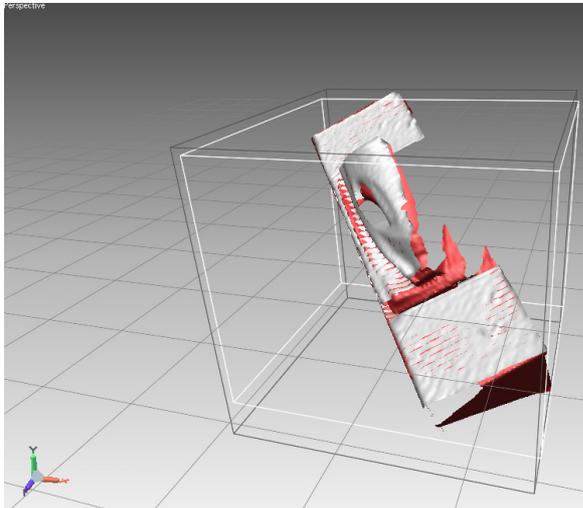
The matrix  $M$  producing the best match of  $V_2$  to  $V_1$  using RapidForm's "Fine Registration" is

$$M = \begin{pmatrix} 0.5828 & 0.4963 & -0.6434 & -820.36 \\ -0.5116 & 0.8393 & 0.18387 & 236.7886 \\ 0.63128 & 0.22205 & 0.74309 & -325.2566 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

where we use the common notation of  $3 \times 3$  rotation submatrix being in the topleft part of  $M$  and the translation specified by the 4th column (length units in mm).



**Figure 2.** Two scanned views of the object shown in Figure 1, referred to as  $V_1$  and  $V_2$ . View  $V_2$  is the one which is roughly aligned with the scanner's  $x$ -axis.



**Figure 3.** View  $V_2$  is matched to  $V_1$  using "Fine Registration" of RapidForm.

The matrices  $F_1$  and  $F_2$  are delivered by the FOB during the digitization of the object.  $V_1$  was obtained while the FOB delivered  $F_1$  and  $V_2$  while the FOB delivered  $F_2$ , where

$$F_1 = \begin{pmatrix} -0.979 & 0.056 & 0.198 & 455.42 \\ 0.196 & -0.026 & 0.98 & -121.41 \\ 0.06 & 0.998 & 0.014 & -120.90 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

and

$$F_2 = \begin{pmatrix} -0.415 & 0.031 & 0.909 & 431.29 \\ 0.909 & -0.04 & 0.415 & -75.44 \\ 0.049 & 0.999 & -0.012 & -121.16 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

The transformation  $N$  from  $V_2$  to  $V_1$ , but now measured in the FOB frame, is  $N = F_1 (F_2)^{-1}$ . We find

$$N = \begin{pmatrix} 0.5873 & -0.8092 & 0.0054 & 141.7320 \\ 0.8098 & 0.5868 & -0.0277 & -429.76 \\ 0.0191 & 0.0209 & 0.9992 & -6.4853 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Now  $M$  and  $N$  are similar matrices, which means that they have the same eigenvalues (not necessarily the same eigenvectors). The most important quantity that  $M$  and  $N$  have in common is the amount of rotation  $\theta$ , defined as

$$\theta = \arccos((w_{11} + w_{22} + w_{33} - 1) / 2),$$

where  $w_{ii}$  refers to matrix diagonal element  $(i, i)$ . We find for  $N$  and  $M$ :

$$\theta_M = 54.37^\circ, \theta_N = 54.08^\circ,$$

indeed very similar values. The interpretation is that the total amount of rotation does not change when measured relative to different coordinate systems. Both the scanned object and FOB's sensor have rotated some 54 degrees be it measured from different frames. According to Poincaré, each rotation matrix also defines the axis about which the rotation occurs as follows:

$$a = \frac{1}{2 \sin \theta} \begin{pmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \\ 0 \end{pmatrix}.$$

For  $M$  and  $N$  we find

$$a_M = \begin{pmatrix} 0.0234 \\ -0.7608 \\ -0.6201 \\ 0 \end{pmatrix} \text{ and } a_N = \begin{pmatrix} 0.0350 \\ 0.0085 \\ 1.0004 \\ 0 \end{pmatrix}.$$

The vectors  $a_M$  and  $a_N$  have unit length approximately. We observe that  $N$  is nearly a rotation about the  $z$ -axis as measured in the coordinate system

of the FOB, whereas  $M$  specifies a rotation about an axis roughly half-way between the negative  $y$ - and  $z$ -directions, as measured in the scanner's frame. This is in accordance with the setup displayed in Fig. 1.

Now we can verify that the axes of rotation are themselves invariant under the rotation, so we expect that  $M a_M = a_M$  and  $N a_N = a_N$ . Numerically we find:

$$M a_M = \begin{pmatrix} 0.03502 \\ -0.76453 \\ -0.61495 \\ 0 \end{pmatrix} \text{ and } N a_N = \begin{pmatrix} 0.019112 \\ 0.005662 \\ 1.0005 \\ 0 \end{pmatrix},$$

which is confirming.

In Figure 3 we noticed that the rotation was clockwise as viewed in the picture, thus counterclockwise about the axis  $a_M$ . So, indeed, the rotation of  $\theta_M = 54.37^\circ$  has the correct (positive) sign.

Until now we only have information about the direction of the rotation axes, not their locations. To find the location of the rotation axis as measured in frame  $S$ , we need to find points  $p_a$  (not vectors) on the line  $l_M$  of rotation. This calls for computing the eigenvectors of  $M$ , as we will do later. The rotation part of  $M$  rotates  $p_a$  by angle  $\theta_M$  about the axis through the origin of the scanner with direction  $a_M$ . By this rotation the point is displaced by the component of vector  $-v_M = (-820.36, 236.7886, -325.2566)^T$  perpendicular to the rotation axis, where  $v_M$  is the translation applied after this rotation, which is the 4th column of matrix  $M$ . By applying the rotation and the translation in sequence, point  $p_a$  remains invariant, as expected for a point on  $l_M$ . The displacement due to rotation,  $|v_M|$  equals to  $2r_M |\sin(\theta_M)|$ , where  $r_M$  is the distance between the line  $l_M$  and the scanner's origin. By definition  $v_M$  should be perpendicular to  $a_M$  (which can be verified by noting that the inner product  $v_M a_M$  vanishes, using the data from  $M$ ). Let us define  $c_M$  as the midpoint of the line of back translation. We still don't know the location of this line. We know that its length and direction are specified by  $v_M$ . We also know that the vector  $c_M$  is perpendicular to both  $a_M$  and to  $v_M$ , or in vector notation

$$c_M = \frac{a_M \times v_M}{|a_M \times v_M|} |c_M|,$$

where  $|c_M| = 1/2 |v_M| / \text{tg}(1/2 \theta_M)$ , from simple planimetry. The point  $d_M$  on the physical line  $l_M$  of rotation closest to the origin of the scanner is

$$d_M = c_M + 1/2 v_M$$

and the point  $d_M - v_M$  is the result of rotating  $d_M$  due to  $M$ . The line of rotation relative to the scanner is thus given by

$$l_M = d_M + \lambda a_M, \lambda \in \mathbb{R}. \quad (3)$$

Similarly, the line of rotation  $l_N$  as observed in the FOB's reference frame is calculated from the data in  $N$  as

$$l_N = d_N + \lambda a_N, \lambda \in \mathbb{R}. \quad (4)$$

Numerically we find

$$l_M = \begin{pmatrix} -19.5 \\ 630.5 \\ -776.0 \\ 1 \end{pmatrix} + \lambda \begin{pmatrix} 0.024 \\ -0.761 \\ -0.620 \\ 0 \end{pmatrix}, \quad l_N = \begin{pmatrix} 491.6 \\ -76.1 \\ -19.0 \\ 1 \end{pmatrix} + \lambda \begin{pmatrix} 0.035 \\ 0.0085 \\ 1.0004 \\ 0 \end{pmatrix}.$$

We recall that  $l_M$  and  $l_N$  represent the same physical axis of rotation, measured in the coordinate systems of  $S$  and  $T$ , respectively. We therefore know the relative placements of  $S$  and  $T$  up to a shift along the axis of rotation. Also the orientation of  $T$  around that axis is undetermined. If we repeat the experiment, where the object is rotated about a different axis (*e.g.* roughly perpendicular to  $l_M$ ), then the placement  ${}^S T$  can be resolved. A calibration procedure could theoretically be organized as follows:

1. Obtain a set of pairs of lines ( $l_M, l_N$ ) where the object displacements should not all have a rotation component about the same axis directions.
2. Create point set  $A$  as a finite collection of points in lines  $l_M$  and, similarly, set  $B$  containing points in lines  $l_N$ .
3. Apply a registration algorithm to the sets  $A$  and  $B$  to find the transformation  $X$  such that the transformed set  $XA$  matches  $B$ . Then  $X$  is the relative placement  ${}^T S$  as used in equation (1).

Obviously, the registration procedure in step 3 is not quite commonly applied since there is no prior knowledge about the correspondence between the points in  $A$  and  $B$ , except for their being on a specific line in space.

Another approach to obtain  $X$  is based on the equality of the eigenvalues of  $M$  and  $N$ . If  $H$  and  $G$  denote the matrices of eigenvectors of  $M$  and  $N$ , respectively and  $\Lambda$  the diagonal eigenvalue matrix (which is equal for  $M$  and  $N$ ), then it holds that  $M = G \Lambda G^{-1}$  and  $N = H \Lambda H^{-1}$  and from equation (1):

$$X H \Lambda H^{-1} X^{-1} = G \Lambda G^{-1},$$

which holds if  $X = G H^{-1}$ .  $X$  is then interpreted as the frame of eigenvectors of  $M$  relative to the frame of eigenvectors of  $N$ . However, since two of the

eigenvalues (of  $M$  as well as of  $N$ ) are equal (to unity), this quantity is not unique. The two eigenvectors with eigenvalue 1 represent the direction of rotation and a point on the line of rotation, where the latter indeed has a non-zero 4th component, as is needed for the homogeneous coordinates of a 3D point. Every linear combination of this point and this direction vector remain invariant under  $M$  (resp  $N$ ), and thus collectively define the line of rotation  $l_M$  and  $l_N$ .

#### 4. METHOD TO DETERMINE ${}^S T$

In the previous section we considered the measurement of a single rotation axis relative to frame  $S$  and relative to frame  $T$ , where these lines are denoted  ${}^S l$  and  ${}^T l$ , respectively. We can construct frames  ${}^S L$  and  ${}^T L$  from the data defining the lines in equations (3) and (4), which we rewrite as

$$\begin{aligned} {}^S l &= {}^S d + \lambda {}^S a \\ {}^T l &= {}^T d + \lambda {}^T a. \end{aligned}$$

Then we define frame  ${}^T L$  as

$${}^T L = \begin{pmatrix} \hat{x} & \hat{y} & {}^T a & {}^T d \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

where  $\hat{y} = {}^T a \times \hat{q}$ , and  $\hat{x} = \hat{y} \times {}^T a$ , where  $\hat{q}$  is any unit vector not parallel to  ${}^T a$ . Recall that  ${}^T a$  is a unit length vector.  ${}^T L$  is a frame with origin  ${}^T d$  and with  $z$ -direction pointing in the direction  ${}^T a$ . When applying the similar construction method to the line  ${}^S l$  we obtain the frame  ${}^S L$ . The matrix  $Y = {}^S L ({}^T L)^{-1}$  has the property that

$$\begin{aligned} Y {}^T L &= {}^S L ({}^T L)^{-1} {}^T L = {}^S L \\ \text{and } Y {}^T l &= {}^S l, \end{aligned}$$

a property that  ${}^S T$  should have too.

However, it can be seen that for any  ${}^T L'$  obtained after shifting  ${}^T L$  over its own  $z$ -axis and/or after rotation about its own  $z$ -direction it holds that  $Y {}^T L' = {}^S L$ . Therefore, if we define

$${}^T L(\delta, \gamma) = {}^T L D_z(\delta) R_z(\gamma),$$

$$\text{where } D_z(\delta) R_z(\gamma) = \begin{pmatrix} \cos \gamma & -\sin \gamma & 0 & 0 \\ \sin \gamma & \cos \gamma & 0 & 0 \\ 0 & 0 & 1 & \delta \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

and we define  $Y(\delta, \gamma) = {}^S L ({}^T L(\delta, \gamma))^{-1}$ , then

$$Y(\delta, \gamma) {}^T l = {}^S l, \text{ for all } \delta \in \mathbb{R}, \gamma \in [0, 2\pi].$$

This confirms that we can retrieve, from data  ${}^S l$  and  ${}^T l$ , the matrix  ${}^S T$  up to transformation  $D(\delta) R_z(\gamma)$ .

A method to find the appropriate  $\delta$  and  $\gamma$  is as follows. Obtain  $n$  measurements  $({}^S l_i, {}^T l_i)$ ,  $i=1, \dots, n$ ,  $n>1$ . From these measurements we can derive  ${}^S a_i, {}^S d_i, {}^T a_i$  and  ${}^T d_i$  and hence the frames  ${}^S L_i$  and  ${}^T L_i$  for  $i=1, \dots, n$ , as described above. Then we search for the pair  $(\delta', \gamma') \in \mathbb{R} \times [0, 2\pi]$  for which holds

$$Y_1(\delta', \gamma') {}^T a_i = {}^S a_i, \quad 2 \leq i \leq n, \quad (5)$$

where  $Y_1((\delta', \gamma') = {}^S L_1 ({}^T L_1((\delta', \gamma')^{-1})$  is the  $Y$ -matrix derived from one particular measurement ( $i=1$  in this case). Here we take all possible transforms derived from  ${}^S L_1$  and  ${}^T L_1$  and test them on the correct transformation of the rotation directions  ${}^T a_2, \dots, {}^T a_n$ .

The test expressed in equation (5) is extended by

$$Y_1(\delta', \gamma') {}^T d_i \in {}^S l_i, \quad 2 \leq i \leq n, \quad (6)$$

since point  ${}^T d_i$  is, by definition, contained in line  ${}^T l_i$  and therefore  $X {}^T d_i$  should be contained in  ${}^S l_i$  as well for any  $i$ . We denote the distance of the point to the line by  $|gl|$ . If equation (5) holds the unit vectors on the left- and right-hand side are equal. We denote their difference as  $|fl|$ .

In Figures 4 and 5 the behavior of  $|fl|$  and  $|gl|$  as function of  $(\delta, \gamma)$  are shown for a simple measurement with  $n=2$ . For the particular experiment (we found  $\delta = -811.7\text{mm}$  and  $\gamma = 239.1^\circ$ , corresponding to

$$X = Y(\delta', \gamma') = \begin{pmatrix} 0.189 & -0.982 & 0.022 & -18.12 \\ -0.662 & -0.144 & -0.735 & 89.21 \\ 0.724 & 0.125 & -0.678 & -1176.54 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Initial numerical experiments indicate that the differences  $|\theta_N - \theta_M|$  (defined in section 2) remain below 0.5 degrees and the differences  $|u_N - u_M|$  remain below 0.5mm.

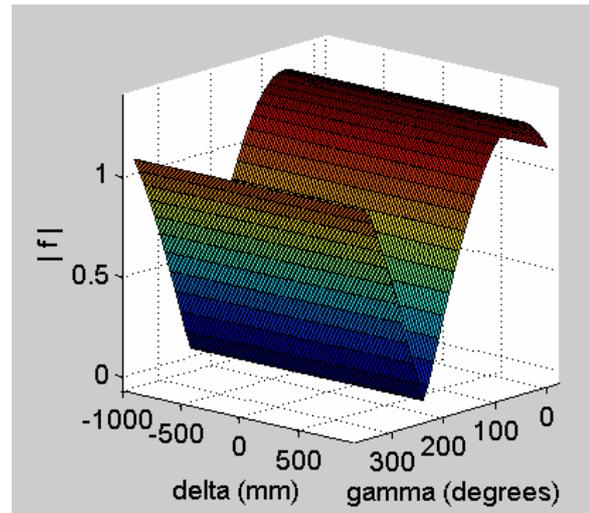


Figure 4. Directional deviation as function of  $(\delta, \gamma)$ .

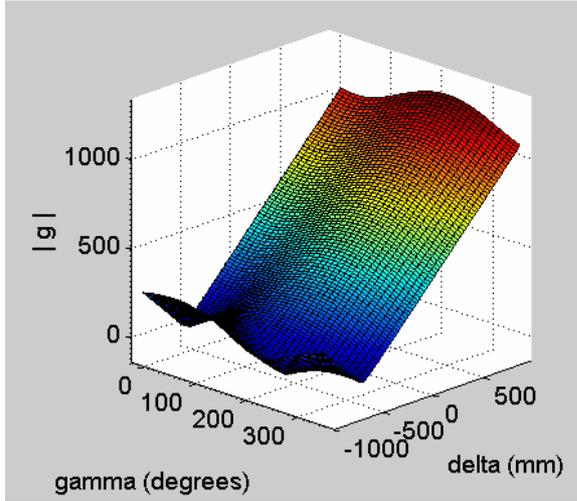


Figure 5. Positional deviation as function of  $(\delta, \gamma)$ .

## 5. SUPPORTED SCAN VIEW REGISTRATION

If we have determined the coordinate system  $X = {}^S T$  of the sensor transmitter's location relative to the scanner's coordinate system we can define a procedure to apply the 6D sensor data in the scan view registration process.

Using the quantity  $X = {}^S T$  we can compute the location of the sensor's local origin relative to the scanner as a function of time:

$${}^S p(t) = {}^S T {}^T p(t),$$

where  ${}^T p(t) = (x(t), y(t), z(t))^T$  represents the position coordinates delivered by the sensor. Similarly, the sensor's orientation relative to  $S$  as a function of  $t$  can be computed. More importantly, if two scan views  $V_j$  and  $V_{j+1}$  need to be registered, the following procedure can be taken during digitization of an object:

1. Take a scan view at time  $t=t_j$  resulting in a point set (or surface mesh)  ${}^S V_j$ .
2. Record the placement  ${}^T F(t_j)$ .
3. Displace the object into a new position and/or orientation, and take a scan view. Let us denote the time at which this scan view is taken as  $t=t_{j+1}$ . The resulting geometric set is  ${}^S V_{j+1}$ .
4. Record the placement  ${}^T F(t_{j+1})$ .
5. Supply the geometric sets  ${}^S V_i$  and  $A_{j+1} {}^S V_{j+1}$  to the scanner's registration software, where matrix  $A_{j+1}$  is defined as

$$A_{j+1} = X {}^T F(t_j) ({}^T F(t_{j+1})^{-1} X^{-1}).$$

Scan views  ${}^S V_j$  and  $A_{j+1} {}^S V_{j+1}$  will approximately match. Depending on the accuracy of the measured sensor placements and on the accuracy of similarity transformation  $X$ , registration of the two scan views can happen without user intervention.

The user can proceed by taking the next scan view at  $t=t_{j+2}$ , which will be pre-positioned using transformation  $A_{j+2}$  as to automate the registration with the previous scans.

If the user decides to change the position of the sensor relative to the scanned object, for example because the sensor occludes a portion of the surface, then the following scan view at  $t=t_{j+n}$  cannot be pre-positioned relative to the previous scans unless the position of the object remains unchanged between  $t_{j+n-1}$  and  $t_{j+n}$ .

The calibration needs to be repeated only when the FOB's magnetic transmitter and the scanner device are moved relative to each other.

## 6. CONCLUSIONS

To make 3D scanning acceptable for inexperienced users, the registration procedure should be simplified. When data is recorded from a 6D sensor attached to the scanned object, the registration algorithm can be augmented. Since no placement data is available from the 3D scanner, but only transformations between scanned views, the calibration of scanner to sensor requires the similarity transform of pairs of transformations to be determined. This condition is different from those of calibration procedures typical for Augmented Reality devices. We verified numerically that based on a single transformation (obtained by applying the registration algorithm), the similarity transform can be obtained up to a translation along the axis of rotation and a rotation about this axis. We proposed a method to resolve the ambiguity by considering two or more registrations. Since two of the eigenvalues of the transformation matrices are equal, the direct solution of the similarity transform in terms of the eigenvectors is not possible with information from a single registration alone. We have found a method to compute the right similarity transform based on multiple measurements. A practical procedure to augment the scanning process has been described.

Work is in progress to determine the accuracy and error sources of the calibration process, as well as the added value of the method in practical applications. Devices different from the FOB will be experimented, since the FOB causes object occlusion and the connection cable presents a hindrance. Wireless sensors or tracking based on markers could be considered as alternatives. The authors would like to thank the reviewers for their helpful comments.

## 7. REFERENCES

- [Asc2006] <http://www.ascension-tech.com>
- [Grasset 2001] R. Grasset, X. Decoret, J-D Gascuel, "Augmented Reality Collaborative Environment Calibration and Interactive Scene Editing", VRIC, Virtual Reality International Conference,.
- [Kato 1999] H. Kato, M. Billinghurst. "Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing", 2nd IEEE and ACM International Workshop on Augmented Reality, pp 85
- [Min2006] <http://www.konicaminolta.com>
- [Rap2006] <http://www.rapidform.com>.
- [Smyth2000] S.N. Smyth and D.R. Wallace (2000), "Towards the synthesis of aesthetic product form". Proc. DETC2000/DTM-14554, ASME, New York.
- [Song2005] Song, Y, Vergeest, JSM, & Bronsvort, WF (2005). Fitting and manipulating freeform shapes using templates Journal of Computing and Information Science in Engineering, 5(2), 86-94.
- [Vergeest2001] .J.S.M. Vergeest, I. Horvath, S. Spanjaard, "A methodology for reusing freeform shape content".Proc. of the 2001 Design Theory and Methodology Conference, DETC'01/DTM-21708, ASME, New York, 2001.
- [Wheeler 1998] Wheeler A. Pretlove J. Parker G. "Augmented reality : calibration of the real and virtual worlds", Telem manipulator and telepresence technologies IV.vol. 3206, pp. 30-40.

# Evaluation of Different Vibration Visualization Modes for Line Tracking

Yuan Wang  
Computer Science School  
University of Manchester  
M13 9PL, Manchester, UK  
wangy@cs.man.ac.uk

Martin J. Turner  
Manchester Computing  
University of Manchester  
M13 9PL, Manchester, UK  
martin.turner@manchester.ac.uk

W. Terry Hewitt  
Manchester Computing  
University of Manchester  
M13 9PL, Manchester, UK  
w.t.hewitt@manchester.ac.uk

## ABSTRACT

Using a vibration magnitude model [Wan06] established for the PHANToM haptic device, we have investigated the usability of different modes within a line tracking task. Three key questions were evaluated: (1) does vibration aid in tracking visible lines? (2) Which of the key vibration modes is preferred, either the vibration online mode where the vibration decreases with the distance from the line, or the vibration offline mode where the vibration increases away from the line? (3) And is linear or model constrained mapping more beneficial to the task?

The experiment results show that although the vibration mode implemented on visible lines is actually more time-consuming than the mode without vibration, the time ratios spent in high accuracy areas with vibration are enhanced. In comparison with the online vibration, the vibration offline mode provides the more accurate performance to both visible and invisible lines. In certain cases, the vibration magnitude model mapping mode is shown to give superior results, however generally it has the similar usability with the linear mapping mode.

## Keywords

Haptic visualization, line tracking, vibration magnitude model, linear vibration mapping, model vibration mapping, vibration online mode, vibration offline mode.

## 1. INTRODUCTION

### 1.1 Visualization

Visualization is a process of presenting data and information that conveys data's meaning, increases information's clarity, and aids user's comprehension through their perception abilities. Geometries are very common in graphical visualization, such as hotspots, flow lines, orthoslices, isosurfaces and so on. These geometries highlight local pattern and global structure features of data fields, and thereby how to effectively represent them becomes critically essential to visualization.

Touch sensation has been increasingly applied into the visualization field, which is normally referred to as haptic visualization. The word "haptic" comes

from the Greek *haptesthai* meaning "to touch", and it can be divided into two types of sensations: kinaesthetic (force, torque) and tactile feedback. Basically kinesthetic feedback is stimulated by bodily movements and tension; while tactile sensation is primarily conveyed by pressure or vibration on the skin (thus here tactile sensation is characterised with vibration). Haptic visualization displays data and information with these haptic sensations.

Tactile sensation has already shown great capability in information display, such as Braille document, however in the haptic visualization field, the research to date has tended to focus more on force and torque, due to their unique characteristics of bidirectional interaction [Bro90] [Iwa93] [Avi96] [Pao98] [Law04]. Owing to its natural characteristics of steering, force has been applied in various pattern recognition schemes, such as special points, lines, and surfaces [Dur98] [Rei03]. So far there has been little discussion about vibration application in visualization, and even less research has addressed how to represent geometries using vibration. The following section reviews recent research into haptic visualization methods pertaining to line geometry.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.  
Copyright UNION Agency – Science Press, Plzen, Czech Republic.

## 1.2 Line Visualization

Lines have drawn much attention of visualization researchers, because of their simple but useful presentation ability for data and information. For instance, in information visualization, we could map month-long temperature data onto a line graph, with the x-axis representing date and the y-axis temperature, and then plot temperatures in different regions using various colourful or textured lines, in order to facilitate comparisons. In scientific visualization, lines are often used as, for example, streamlines portraying vectors in a flow field. Moreover other attributes, such as vector magnitude, can be mapped via colour on the line. However, many lines presented at the same time can result in visual clutter, and cause serious confusion. Under this circumstance, is there any other way to aid in displaying these lines? Haptic sensation has been revealed as an alternative. In this section, haptic line visualization methods are introduced, according to two main categories: implicit line mapping and explicit line mapping.

### 1.2.1 Implicit Line

Implicit line mapping is used to refer to the method, with which no pre-integrated line exists, but the movement of the user intuitively follows a line. Durbeck et al. [Dur98] first engaged in highlighting flow lines with haptic display, by directly mapping vectors to forces. This method is incredibly effective and straightforward, as the stylus instinctively drags the user along the lines. The “Relative Drag” method proposed by Pao et al [Pao98] is very similar to Durbeck’s method, although the user’s velocity was also encoded into the force rendering.

A few years later, Lawrence and Pao et al. [Law04] developed different implicit line mapping methods for several applications. They recreated the implicit physical phenomenon in a magnetic field, by simulating Lorentz force. In a vortex field, they applied a force  $\vec{F} = g\vec{\omega} \times \vec{v}$  ( $g$  is a gain parameter,  $\vec{\omega}$  is local vorticity vector and defined as the curl of velocity  $\vec{v}$ ) to the stylus, and this force tended to pull the user towards the vortex core. Nonetheless it is evident that only force, owing to its navigation characteristics, can function in implicit mapping.

### 1.2.2 Explicit Line

Several other studies investigating line visualization have been carried out on pre-generated lines, which fall within the scope of explicit line mapping. Torque, damping force [Pao98], and gravity force weighted by several lines [Rei03] have been used to constrain the user to the line. Fritz and Barner [Fritz96] designed a weighting function, which

allows a snap-to-grid force that pulls the stylus to the cylinder axis (which is the graph line) and then enables the user to move freely along the line.

In addition to forces and torques, tactile sensation has also demonstrated its usability for explicit line graph display. In 2002, Roberts et al. [Rob02] published a paper, in which they turned a graph with lines into a greyscale image, and then mapped different grey-values to heights of the surface for exploration with a point. Since there was no other information in the graph besides the lines, a haptic bump-map with bumps and ridges was generated. Friction and noise texture were proposed to be added to provide additional information, of say another variable.

In an attempt to make partially sighted or blind people easily explore line graphs, Yu et al. [Yu00] [Yu01] constructed V-shaped grooves on the surface to simulate the graphs with the PHANToM device. They [Yu02] also enabled automatic online haptic line construction using a Logitech WingMan Force Feedback Mouse. To avoid the confusion inflicted by having multiple lines, different frictions were applied to each of them. When line graphs become complex, they may cross each other. In this situation, the constrained force mapping, with an extra tactile feedback alarm to inform the user where the intersection points are [Rob02], would be more suitable than using an exclusive tactile display.

The assistance of force and torque to line tracking are well documented, and friction, texture and other tactile sensations have also been widely applied. The effects of vibration solely to this task, however, are less well understood. Does vibration have a positive influence on line tracking task? Are there alternative ways to apply vibration to this task?

## 1.3 Paper Outline

In this research, we will mainly discuss vibration sensation, and examine whether vibration can help the user in the line tracking task. The main areas investigated are:

- (1) Does vibration, as an extra information channel, assist the user in tracking the visible line?
- (2) Which vibration mode is more helpful, **VibOnline** mode (the strongest vibration on the line, and the farther the distance to the line, the weaker the vibration) or **VibOffline** mode (no vibration on the line, and the farther the distance from the line, the stronger the vibration)?
- (3) Which changing pattern of vibration is more suitable, the Linear Vibration mapping (**LV**) or the mapping according to the Model [Wan06] (**MV**)?

The model mentioned here was built to provide a magnitude mapping standard associated with both vibration frequency and amplitude parameters.

Thus this paper first gives a brief introduction to the vibration magnitude model in Section 2, including the basic psychophysical experiments and the mathematical construction. Section 3 describes the line tracking experiments, with details in stimuli, design and result analysis. Finally concluding remarks are given in Section 4.

## 2. VIBRATION MAGNITUDE MODEL

The vibration magnitude model [Wan06] provides a basic mapping criterion for haptic visualization, and enables users to extract various vibration mapping tables. A PHANToM Desktop device (Figure 1), made by SensAble Technologies Inc., was adopted in the model construction experiments. As a single pointing handheld device, it allows 6 degree of freedom (DOF) input and provides 3 DOF output.



Figure 1. PHANToM Desktop Device.

Vibration was produced by applying a sinusoidal vertical force along the y-axis, updating at 1K Hz. As a result of the device limitation, the maximum frequency for the test was set as 200 Hz. During the following experiments, the participants were asked to hold the haptic stylus like a pen, and position the probe at the centre of the haptic scene.

### 2.1 Vibration Experiments

This vibration magnitude model was built upon two psychophysical experiments: magnitude estimation and intramodal matching experiments.

#### 2.1.1 Magnitude Estimation Experiment

The first one adopted the magnitude estimation method initially put forward by Stevens [Stevens75]. In the experiment, the subject perceived a series of stimuli, and assigned numbers to match their levels (e.g., intensity). We sampled 16 frequencies from 20 to 200 Hz, and set 6 different amplitudes as testing stimuli for each frequency. A total of 23 subjects took part. All the results obeyed the power law, i.e., the relationship between sensation magnitude and vibration stimuli amplitude is a power function. And therefore 16 power functions with different power exponents, respectively reflecting the relationship between sensation and amplitude for different frequencies, were obtained

#### 2.1.2 Intramodal Matching Experiment

The second experiment was an intramodal matching test, which conducted the user to answer the question such as “What amplitude of vibration at 60 Hz feels as strong as a given vibration at 120 Hz?” Thus the user adjusted the parameter of the comparison stimulus to match the standard stimulus, in order to give an appropriate reply. The vibration with 100 Hz frequency, and 0.06 and 0.1 N force amplitude were individually set as standards, and 22 subjects participated. Finally two equal intensity isolines with the frequency range from 20 to 200 Hz were acquired, which manifest the relation between sensation and frequency.

## 2.2 Vibration Magnitude Model

With the results above, a vibration magnitude model was constructed creating a 3D surface. We first standardised all 16 power functions using two isolines, sampled 9 points on each of them, and then fitted a B-Spline surface using these  $9 \times 16$  control points, shown in Figure 2.

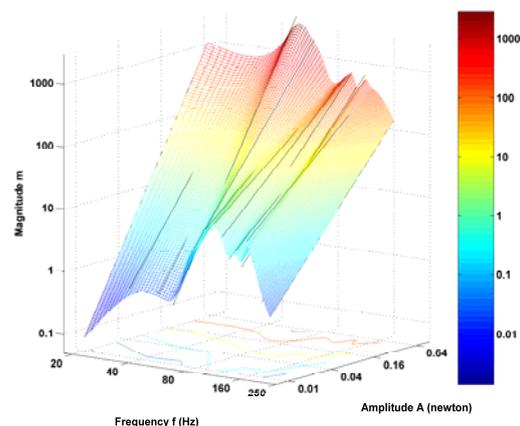


Figure 2. Vibration Magnitude Model.

Based upon this model surface, we can tabulate any number of points, pick out certain points with consideration of their magnitudes, and create a vibration look-up mapping table. This model laid out the foundation for the vibration mapping in the following line tracking experiment. Figure 2 shows that the model has the lowest valley in the frequency range roughly from 70 Hz to 120 Hz (the valley frequency changes as the amplitude changes). Thus in line tracking, we did not extract the mapping table from the whole frequency range of the model, but specified the frequency varying between 80 and 200 Hz and amplitude increasing from 0.06 to 0.18 N. The reasons and details about the vibration mapping table are given in the next section.

### 3. LINE TRACKING EXPERIMENT

In this section, we investigate vibration in the line tracking experiment, therefore only vibration was employed as the haptic guidance.

#### 3.1 Design

The study used the PHANTOM Desktop device and a 19-inch display monitor with a resolution of 1280×1024, respectively for haptic and graphic display. The virtual environment was constructed using the GHOST SDK [Gho06] and the embedded OpenGL package, haptically updated at 1000 Hz and graphically at a rate of 67 frames per second. In the experiment, a 2D line was placed on a vertical background surface. The space close to the line was partitioned into eight-level vibration areas (Figure 3), where vibration changed discretely depending on the distance to the line. The experiment task for the participant was to track the trial line according to the perceived vibration hint and/or visual feedback.

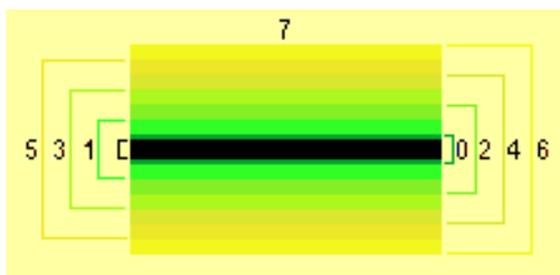


Figure 3. The Eight-level Areas around the Line.

The experiment had a 2×4 within-subjects factorial design. The factors were vibration mode (VibOnline or VibOffline), and type of tracking guidance (Linear mapping Vibration (LV), Model mapping Vibration (MV), Vision only (V), Vision + Model mapping Vibration (V+MV)). How VibOnline and VibOffline modes functioned is explained in Section 1.3.

### 3.2 Stimuli

There were four lines in total. And each of them was 2.0 mm wide, 350 mm long, and contained three different rectangular grooves, which can be upward or downward. The two of them were the upside-down replicas of the other two. Thus the two lines respectively had two narrow grooves (20 mm width) and one wide groove (40 mm width), while each of others had one narrow groove and two wide grooves. Figure 4 gives an illustration of a trial line's pattern.

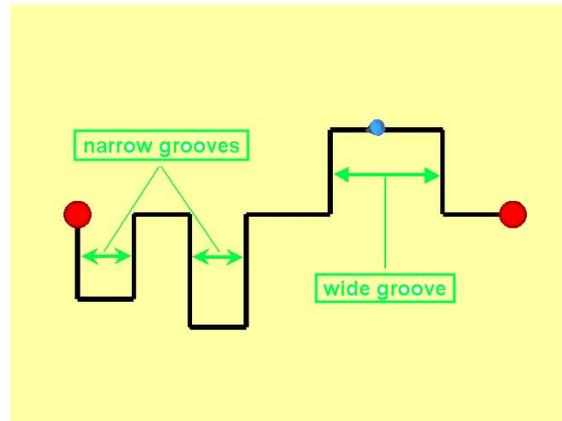


Figure 4. A Trial Line in the Line Tracking Experiment.

The vibration varied according to the distance using different mapping methods (LV or MV). Table 1 shows the eight-level stimuli of VibOffline mode for LV and MV. The mappings in the reverse order were used for VibOnline mode. With the consideration of the vibration magnitude model, the frequency range here was chosen from 200 to 80 Hz, which promised the comparatively monotonic sensation variation for even LV mapping.

Level	Distance from line (mm)	LV		MV	
		Fre (Hz)	Amp (N)	Fre (HZ)	Amp (N)
0	$0 \leq d < 1.5$	200	0.0	200	0.0
1	$1.5 \leq d < 3$	200	0.06	200	0.06
2	$3 \leq d < 4.5$	180	0.08	186.5	0.074
3	$4.5 \leq d < 6$	160	0.1	173	0.091
4	$6 \leq d < 7.5$	140	0.12	155	0.091
5	$7.5 \leq d < 9$	120	0.14	137	0.115
6	$9 \leq d < 10.5$	100	0.16	114.5	0.165
7	$10.5 \leq d$	80	0.18	80	0.18

Table 1. Vibration Offline Levels Table for the Linear and Model Maps

### 3.3 Procedure

Four female and five male subjects, aged between 22 and 35, took part. To start a trial, the participant positioned the cursor (represented by a blue cone) in contact with the left sphere, and pressed the button on the stylus. Then he moved the cursor to track the line through vibration, or vision, or both, until reaching the right sphere. At this time, the participant pressed the button again, and the trial was ended.

Each participant completed two practice trials and then 16 test trials. VibOnline and VibOffline modes were mixed during the experiment, but each of them was always applied into two trials in a row, the next two trials adopted the other mode. Except for this limitation, all the trials were presented at random. Participants were explicitly informed which vibration mode (VibOnline, VibOffline) would be used and whether vision would be provided. After each trial, the participant was asked to rate on a seven-level scale how difficult they thought the particular trial was. When all the trials were finished, they were required to answer the last question: “which vibration mode is more comfortable for completing the task? The options are: vibration Online, vibration Offline or both the same.”

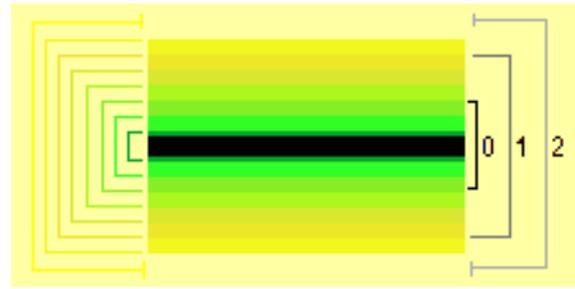
### 3.4 Results

During each trial of the experiment, we recorded the total movement time, the percentage of the time (time percent) spent in each level area, and the number of entries (entry number) into each level area (except for 0-level) once moving from the lower level area to the current level one. The latter two items describe the accuracy and error rate during the subject tried to complete the task. To comprehensively assess influences various factors exerted on the task, the comparisons were made from the four aspects:

- (1) All factor comparison (LV, MV, V, & V+MV; VibOnline & VibOffline)
- (2) Vision-excluded factor comparison (LV & MV ; VibOnline & VibOffline)
- (3) Vision-based factor comparison (V & V+MV)
- (4) Vision-vibration-based factor comparison (Vib-Online & VibOffline)

Through the initial observation on the data, we found Areas 0, 1 and 2 were predominantly involved, under vision-based guidance, thus we set the following three regions for further investigation (Figure 5).

- (1) Accurate region 0: includes Areas 0, 1 and 2
- (2) Close region 1: includes Areas 3, 4 and 5
- (3) Deviation region 2: Areas 6 and 7

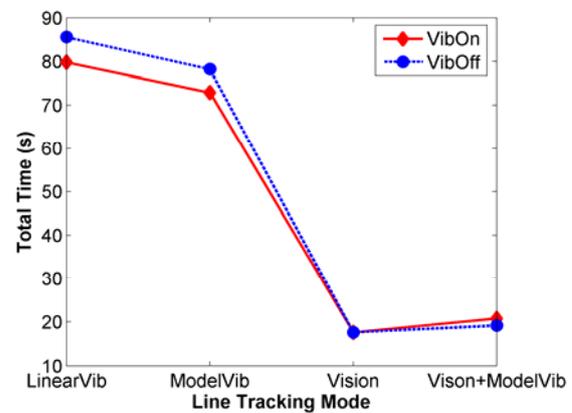


**Figure 5. The Three-level Regions around the Line.**

And actually for each comparison, we differently analysed the results collected in areas or regions.

#### 3.4.1 All Factor Comparison

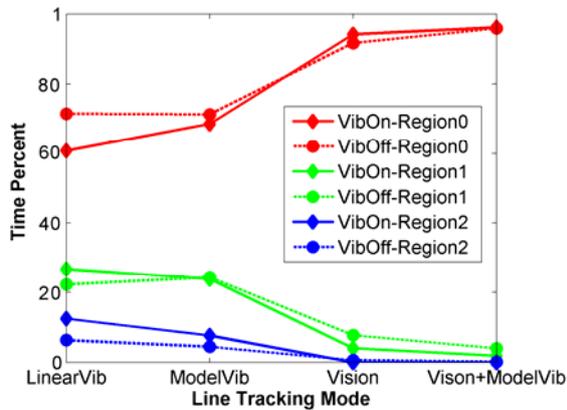
The mean values of total time under the various modes and tracking guidance conditions are shown in Figure 6. As expected, tracking guidance performs a considerable influence on the total time results ( $F_{3,24} = 51.992, p < 0.01$ ), since the vision-based guidance leads to much shorter movement time than the vision-excluded one. Moreover, from the data in Figure 6, under the guidance of only vibration, the model mapping mode aids in accomplishing the task a bit faster than the linear mapping mode does. However unexpectedly, vibration, as an extra guidance hint, fails to speed up tracking the visible lines, but slightly prolongs it instead. More details about these are given in other factor comparisons.



**Figure 6. Mean Values of Total Time.**

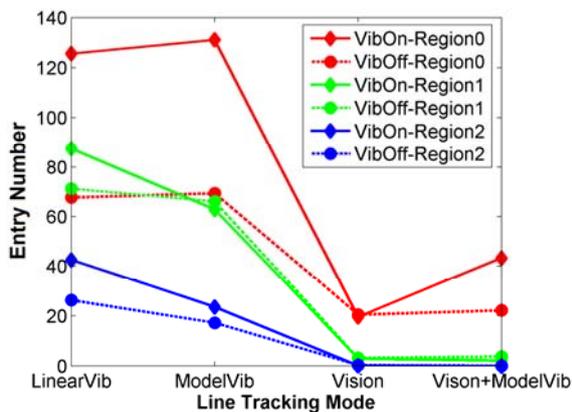
In this comparison, we used the region standard to analyse the result. The movement time percent in each region under different guidance conditions are quite different (see Figure 7). A repeated measure ANOVA confirms this, yielding a significant effect for each region (Region 0:  $F_{3,24} = 48.947, p < 0.01$ ; Region 1:  $F_{3,24} = 74.133, p < 0.01$ ; Region 2:  $F_{3,24} = 9.9144, p < 0.01$ ). The results under the

vision-based guidances are dominantly located in Region 0, while those under the vision-excluded guidances are relatively averagely but decreasingly distributed across Regions 0, 1 to 2. This distinction indicates that vision predominates over haptic vibration. However the online or offline mode does not show a significant effect on performance.



**Figure 7. Time Percent Mean Values Spent in Regions 0, 1 and 2.**

The entry number means in Figure 8 similarly reinforce that vision is an advantage over vibration (Region 0:  $F_{3,24} = 53.61$ ,  $p < 0.01$ ; Region 1:  $F_{3,24} = 42.70$ ,  $p < 0.01$ ; Region 2:  $F_{3,24} = 11.51$ ,  $p < 0.01$ ). In addition, we found that VibOffline mode showed the better performance than VibOnline mode. However since vision only guidance does not take into account vibration, the ANOVA result here may underestimate the effect of Online/Offline. Therefore under the following comparisons, we make a thorough analysis of these modes and conditions.



**Figure 8. Entry Number Mean Values in Regions 0, 1 and 2.**

### 3.4.2 Vision-excluded Factor Comparison

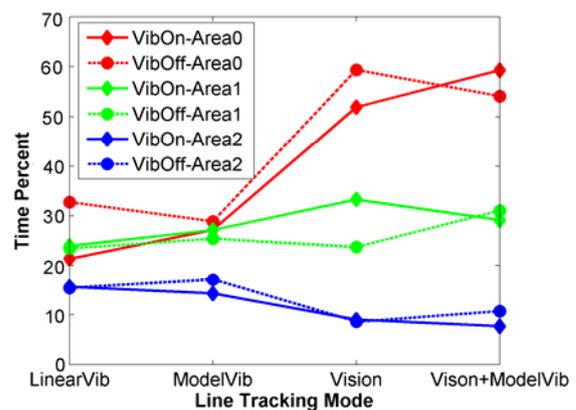
Referring to Figure 6, as mentioned before, we can see that ModelVib takes shorter time than LinearVib,

and this indicates the model mapping's superiority over linear mapping. However their difference is not significant, probably stemming from the intentionally chosen limited vibration range (80—200 Hz rather than 20—200 Hz). Figure 6 also shows that VibOnline results in faster movement than VibOffline. The most likely cause of this distinction is that with the VibOffline mode the subject tried to seek the perfect position without vibration at all, and with the VibOnline mode the subject tended to go along with the hint of a strong vibration. The subconscious perfection seeking under the VibOffline mode, therefore, appears to make it a longer task, although no statistical significance is associated with.

The vibration-excluded factor comparison analysed the results collected from regions. VibOffline mode spent a higher time percentage in the accurate region 0, and the lower percent in the other two regions than VibOnline mode (Figure 7). The VibOffline mode's accuracy advantage over the VibOnline mode has been statistically manifested by the entry number results (Region 0:  $F_{1,8} = 10.583$ ,  $p < 0.025$ ). The VibOffline mode proceeded more accurately.

### 3.4.3 Vision-based Factor Comparison

Comparing the data in Figure 6, we can see that any extra vibration condition has the minimal impact on the results of visible line tracking, owing to the preponderance of vision over haptic vibration. However we surprisingly found that vibration protracted the tracking task, instead of shortening the movement time like force did.



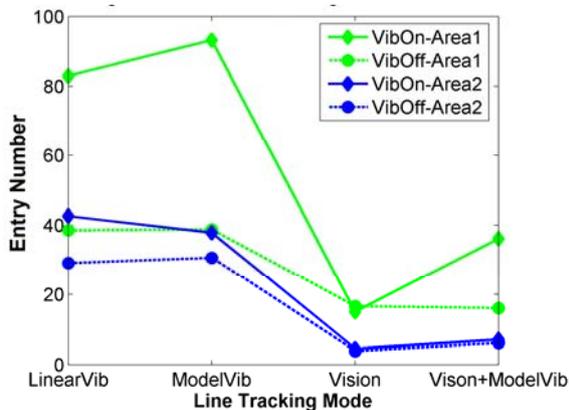
**Figure 10. Time Percent Mean Values Spent in Areas 0, 1 and 2.**

Due to the precise results from vision-based modes, we only compare the results in Areas 0, 1 and 2. Data from Figure 10 reveal that the time percents under V and V+MV guidances in each area are fairly similar, no evident pattern existing, and a repeated measures of ANOVA confirms this. However, referring to

Figure 7, we can see that to the extent of regions, vibration slightly but uniformly augments the percentage of time spent in accurate region, and diminishes the time ratio in the far regions. Thus although vision plus vibration guidance is more time consuming, this multimodal interaction improves the task accuracy, regardless of an absence of significant difference between V and V+MV modes.

On the other hand, the entry number result shows some interesting difference (Figure 11). Vibration, as an additional guidance hint, unexpectedly increases the error rate during tracking visible lines (Area 1:  $F_{1,8} = 5.53, p < 0.05$ ; Area 2:  $F_{1,8} = 11.29, p < 0.05$ ).

This possibly results from the participant's subconsciousness of exploring different vibrations under V+MV Online mode and reconfirming whether he is in the accurate area under V+MV Offline mode. However in the far regions, the entry number results under V and V+MV guidances are roughly identical (see Figure 8).



**Figure 11. Entry Number Mean Values Spent in Areas 1 and 2.**

### 3.4.4 V+VibOnline & V+VibOffline Factor Comparison

ANOVA outcomes reveal that the total time results under these two modes make no difference, and no significant distinction existed in time percent either. However Offline mode does have an advantage over Online mode (Area 1:  $F_{1,16} = 4.9815, p < 0.05$ ).

### 3.4.4 Comparison Summary

All the comparisons evaluating the vibration-aided line tracking can be summarised from three aspects:

1. Line tracking under the vision only guidance is definitely superior to that exclusively in aid of vibration. However in the multimodal tracking task, vibration, as an extra feature, magnifies some amount of time percentage in the accurate region. This proves the possibility to apply vibration in

multimodal line tracking, and even to convey other parameter of line (such as magnitude on streamline) with vibration.

2. In contrast to the VibOnline mode, the VibOffline mode does give rise to more accurate results, no matter whether under vision-excluded guidance or in multimodal interaction. This is consistent with the questionnaire result that 7 out of 9 subjects chose that "VibOffline mode is more comfortable". Nonetheless it is of great interest to further investigate this issue, especially when the pattern of the line becomes complex, or when the number of the lines increases. Furthermore, although VibOnline mode is not the better way to track line, it is more practically extendable, such as mapping other variable of the line to vibration. However the inferiority of VibOnline mode does give us a potential warning in this extension.

3. In certain cases, the linear mapping of vibration with the appropriate range has similar usability when compared with the model mapping. Regardless of this, the vibration magnitude model will exhibit its superiority if the ranges of both frequency and amplitude are extended. Therefore the vibration mapping based on the magnitude model is strongly recommended.

## 4. CONCLUSIONS AND FUTURE WORK

This paper aims to examine how to effectively apply vibration in one specific visualization task, line tracking. Based upon the fundamental vibration magnitude model, vibration has been applied in different ways. The implementation of this experiment accumulates the practical experience for our motivation. How to employ vibration in more complicated applications to facilitate the user interaction will be an important direction of the future research.

During the experiment, the participants reported some kind of jump or jerk occurred occasionally. This phenomenon is most likely caused by the sudden change of the vibration, and the solution to this problem will be interpolation. Even though we have not addressed this issue in detail here, it is necessary to reasonably implement interpolation to smooth data gaps within real applications.

## 5. ACKNOWLEDGMENTS

The authors would like to thank the colleagues to volunteer themselves to take part in the experiments, and also appreciate James Perrin for his technical support. One of us, Yuan Wang, acknowledges the financial support of the University.

## 6. REFERENCES

- [Avi96] Avila R.S. and Sobierajski L.M. A Haptic Interaction Method for Volume Visualization. In Proceedings of the 7<sup>th</sup> Conference on Visualization '96, San Francisco, CA, USA, IEEE CS Press, pp.197-204, 1996.
- [Bro90] Brooks J.F.P, Ouh-Young M., Batter J.J. and Kilpatrick P.J. Project GROPE – Haptic Displays for Scientific Visualization. In Proceedings of the 17<sup>th</sup> Annual Conference on Computer Graphics and Interactive Techniques, Dallas, TX, USA, ACM Press, pp.177-185, 1990.
- [Dur98] Durbeck L.J.K., Macias N.J., Weinstein D.M., Johnson C.R. and Hollerbach J.M., SCIRun Haptic Display for Scientific Visualization. In Proceedings of the 3<sup>rd</sup> Phantom Users Group Workshop (PUG 98), Cambridge, MA, USA, October, 1998.
- [Fri96] Fritz J.P. and Barner K.E. Design of a Haptic Graphing System, In Proceedings of the 19<sup>th</sup> Rehabilitation Engineering & Assistive Technology Society of North America Conference, Salt Lake City, UT, USA, pp. 158-160, June, 1996.
- [Gho06] GHOST SDK Software document, at [http://www.3dtouch.net/products/phantom\\_ghost/ghost.asp](http://www.3dtouch.net/products/phantom_ghost/ghost.asp)
- [Iwa93] Iwata H. and Noma H. Volume Haptization. In Proceedings of IEEE Virtual Reality Annual International Symposium, Seattle, WA, USA, pp.16-23, September 1993.
- [Law00] Lawrence D.A., Lee C.D. and Pao L.Y. Shock and Vortex Visualization Using a Combined Visual/Haptic Interface. In Proceedings of IEEE Conference on Visualization and Computer Graphics, Salt Lake City, UT, USA, pp. 131-137, October 2000.
- [Law04] Lawrence D.A., Pao L.Y., Lee C.D. and Novoselov R.Y. Synergistic Visual/Haptic Rendering Modes for Scientific Visualization. IEEE Computer Graphics and Applications, No.24, pp. 22-30, November/December 2004.
- [Pao98] Pao L.Y. and Lawrence D.A. Synergistic Visual/Haptic Computer Interfaces. In Proceedings of Japan/USA/Vietnam Workshop on Research and Education in Systems, Computation, and Control Engineering, Hanoi, Vietnam, pp. 155-162, May 1998.
- [Rei03] Van Reimersdahl T., Bley F, Kuhlen T. and Bischof C.H. Haptic Rendering Techniques for the Interactive Exploration of CFD Datasets in Virtual Environments. Proceedings of the 7<sup>th</sup> International Immersive Projection Technologies Workshop and the 9<sup>th</sup> Eurographics Workshop on Virtual Environments, pp. 241-246, Zurich, Switzerland, May, 2003.
- [Rob02] Roberts J.C., Franklin K. and Cullinane J. Virtual Haptic Exploratory Visualization of Line Graphs and Charts. In Proceedings of the Engineering Reality of Virtual Reality, pp. 401-410, January 2002.
- [Rob05] Roberts J.C. and Franklin K. Haptic Glyphs (Hlyphs) – Structured Haptic Objects for Haptic Visualization. In Proceedings of WorldHaptics – the 1<sup>st</sup> Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, IEEE CS Press, Pisa, Italy, pp. 369-374, March 2005.
- [Yu00] Yu W., Ramloll R. and Brewster S. Haptic Glyphs for Blind Computer Users. In 1<sup>st</sup> International Workshop on Haptic Human-Computer Interaction, pp. 102-107, Glasgow, UK, 2000.
- [Yu01] Yu W., Ramloll R., Brewster S. and Riedel B. Exploring Computer-Generated Line Graphs Through Virtual Touch. In Proceedings of International Symposium on Signal Processing and its Applications (ISSPA), IEEE CS Press, pp. 72-75, Kuala Lumpur, Malaysia, pp. 13-16 August, 2001.
- [Yu02] Yu W., Cheung K. and Brewster S. Automatic Online Haptic Graph Construction. In Proceedings of EuroHaptics, Edinburgh, UK, pp. 128-133, July, 2002.
- [Wan06] Wang Y., Turner M.J. and Hewitt W.T. A Haptic Visualization Mapping Model – Magnitude Model of Sinusoidal Vibration. In Proceedings of EuroHaptics, Paris, France, pp. 345-350, July 2006.

# A Crowd Simulation Using Individual-Knowledge-Merge based Path Construction and Smoothed Particle Hydrodynamics

Weerawat Tantisiriwat  
Dept. of Computer Engineering  
Faculty of Engineering  
Chulalongkorn University  
Bangkok 10330, THAILAND.  
weerawat.tan@gmail.com

Arisara Sumleeon  
Dept. of Computer Engineering  
Faculty of Engineering  
Chulalongkorn University  
Bangkok 10330, THAILAND.  
arizx@hotmail.com

Pizzanu Kanongchaiyos  
Dept. of Computer Engineering  
Faculty of Engineering  
Chulalongkorn University  
Bangkok 10330, THAILAND.  
pizzanu@cp.eng.chula.ac.th

## ABSTRACT

Previous researches on crowd simulation are often based on pre-computed path planning to reduce computational time; however, the pattern of the crowd locomotion is usually unrealistic and the simulation itself is also inflexible. This paper presents a novel technique to model an interactive time crowd simulation by using automatic path construction and Smoothed Particle Hydrodynamics (SPH) in order that each individual can automatically search for the destination without path pre-computation. Two alternating stages approaches are used in our simulation. Firstly, the environment and crowd are set up, and crowd is then moved in accordance with SPH in order to achieve smooth locomotion without fixed pattern. Each agent from a group of crowd can create a map that contains the knowledge of local environment that it gains from perceptions. The local path can be extended and shared corresponding to the new environment or knowledge sharing with other agents. An automatic path determination is done in the next stage by using the potential field. The result from path selection is then used to set the direction of external force in SPH model. By using our method, natural crowd locomotion under a variety of conditions, such as forming and separating lane, obstacles avoidance and escape from unknown area can be represented in interactive time.

## Keywords

Crowd simulation, Particle methods, fluid dynamics, Smoothed Particle Hydrodynamics

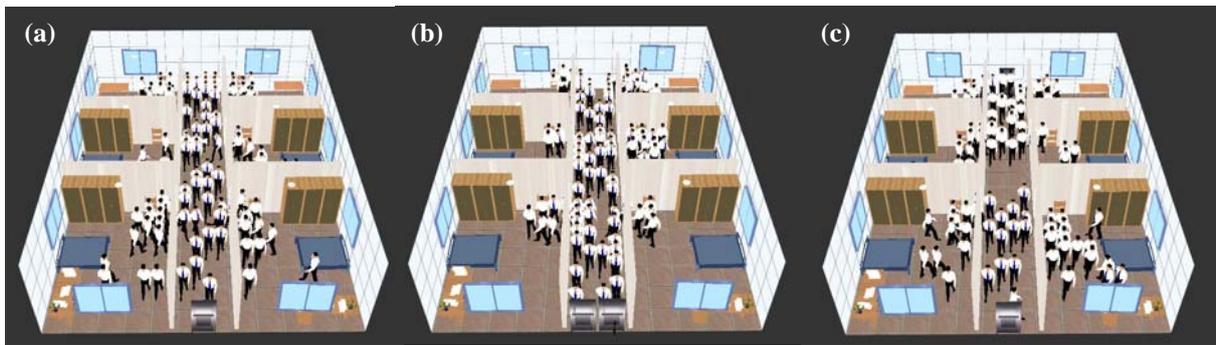
## 1. INTRODUCTION

With the advance in computer technology, it is not surprising that sometimes we can not figure out between what is real and what is computer-generated. Crowd simulation is one of the most popular among computer graphics researchers. Visual realism of crowd simulation involves more than model for decision and evaluation of environment for automatic interactivity of individual since it also has to avoid collision while moving in local direction and matching velocity of neighboring individual. It is interesting and yet not easy to model a crowd simulation that has natural crowd locomotion. This research proposes a crowd

simulation that can realistically model path searching behavior of crowd when it is in unknown place by using automatic path construction, which is modeled by imitating decision behavior of human. Local knowledge of each individual is going to be evaluated in order to find path. Crowd locomotion can be described using fluid dynamics which can naturally simulate crowd motion.

Many researches have proposed various techniques based on pre-computed path for guiding each individual to the destination, for example, Probabilistic Roadmap which uses the construction of all possible path in the environment and A\* which uses heuristic function for searching path to the destination. In addition, potential field is another method that is used to guide crowd locomotion toward the destination. Despite the fact that path planning can make crowd motion smoother, the decision behavior is not modeled naturally since each individual has knowledge of overall path in environment. Imagine if someone can escape from the unknown place where he has never been to, this is not possible since human gain knowledge from

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.  
Copyright UNION Agency – Science Press, Plzen, Czech Republic.



**Figure 1. Path selection of crowd simulation (a) one destination, (b) two closed destination, (c) two separated destination**

experience, which is different from person to person, and then that knowledge is used to select path. As a result, the assumption of global knowledge about path in the environment presented in previous works is not correctly model human decision process.

Agent-based approach is widely used to model crowd. It suits well with human behavior model since thought, decision and locomotion of each individual can be freely determined. However, it is time-consuming when applied to crowd simulation. For this reason, fluid dynamics is used to describe crowd locomotion. This method can model visual plausibly large crowd simulation. Without using agent in modeling, crowd can only move to its assigned direction and the model have no decision behavior.

For this reason, we propose a crowd simulation that coupling fluid dynamics with Agent-based approach, which is used in modeling human decision, path searching and path construction from local knowledge to create characteristic crowd locomotion. We can animate crowd simulation that has more visual realism in path determination and each individual's decision. Our model can simulate smooth crowd locomotion under various environments and conditions.

This paper is organized as follows: in Section 2 some related works are presented, in Section 3 we describe the path construction approach. Locomotion behavior is proposed in Section 4. The implementation is described in Section 5. Section 6 presents obtained result, while Section 7 is conclusions and future works.

## 2. RELATED WORKS

Since 1987, Reynolds has proposed Boids-model that described behavior of unit in the group by using specific local rule for assigning each unit behavior [Rey87b]. It was developed thereafter to model automatic interactivity [Rey99c] which was based on the idea that unit can move toward its assigned destination while handling collision avoidance from surrounding units and obstacles in the environment and locating near its neighboring units to form group behavior. This work has been developed in many

works to increase the realism in various aspects. Social potential field [Rie95a] describes the interaction force between each unit of group. It can be repulsive when each unit reaches the limited distance. However, it was not model crowd realistically. Braun has proposed crowd simulation that model individual behavior [Bra03b]. Each individual behavior is modeled by using various parameters. This research based on Herbing model [Hel00a] that suits with crowd simulation in emergency. It was improved to increase more realism by [Lat04a] which has proposed emergency avoidance force and force that try to help the victims. Wei has proposed Autonomous Pedestrians [Wei05a] that focus on modeling interactive behavior of crowd to the environment in its eyes sight. Hughes has proposed a crowd simulation using fluid dynamics to describe crowd locomotion [Hug03b]. Crowd is viewed as continuous dense particle. This work can animate impressive crowd locomotion that is frequently seen. Recently, Treuille has developed 3D crowd simulation based on previously stated work [Tre06a]. Massive amount of crowd can be simulated in interactive time by calculating equations of fluid motion using Eulerian scheme. This work can generate smooth crowd locomotion but the detailed behavior of each individual is lacked and the computation time is based on control volume.

However, the model of crowd behavior is not enough for realistic crowd simulation. Pre-computation of path is one of the factors that make crowd animation looks more plausibly. Various techniques have widely been used. One of them is path construction using grid table in A\* [Rus94d]. Using this method can easily bring unit to the destination; however, the resulting animation is not plausible especially in vehicles motion around the corner. Probabilistic Roadmap (PRM) [San02a] uses random placing of points onto the environment and connecting these points using lines to obtain the main connected path. Bayazit [Bay04a] is coupling PRM with flocking for flocking simulation. However, the efficiency of the constructed path depends on the number of

randomly-placed points. Using potential field to construct the path of crowd is one alternative that can generate smooth crowd motion. Its principle lets crowd moves toward the direction that has less potential value which can specify the factors that affect the potential value as Treuille has presented in continuum crowds [Tre06a].

Although pre-computed path makes crowd locomotion look smoother, it is still lack of a natural decision behavior since each individual is assumed to have same level of knowledge. It is not possible since human gain knowledge from experiences and uses these experiences to select different paths. This can be seen in the work of Murakami has proposed which stated about using agents to bring the victims to the exit [Mur02b]. Crowd has unordered movements that are resulted from individual perception in its area. Agents can choose the corrected path to the exit using less time while other people who have to randomly search for the exit itself or they can escape faster when they were told by agents.

We propose the crowd simulation using Individual-Knowledge-Merge based Path Construction and Smoothed Particle Hydrodynamics which integrates decision model using individual knowledge and can be shared and merged to create global path without the need of pre-computation of path. As a result, crowd can have more natural locomotion than previous methods. SPH is pure Lagrangian method; it means that we can explicitly specify the rules to individual. In our system, we model crowd volume as a set of particles which each of these represents each individual composing crowd.

### 3. PATH CONSTRUCTION

In this section, we will discuss about the path construction of crowd so that it can move to any positions in the environment. Path is constructed based on local knowledge of each individual, which is different from individual to individual and results from each individual's experience. It means that if one individual have global knowledge, it can construct the path direct to the destination without randomly move through the environment. We will describe the design of knowledge construction imitated from human perception by classifying it as perception from vision and from communication. We then described about knowledge analysis to construct a path of each individual. The model of optimized path decision which is derived from various factor and knowledge sharing of each individual that is used to construct global path of the environment is described in last two subsections.

## 3.1 Perception

Perception is the starting point in development of human abilities. The most obvious and most used is vision and communication and these abilities are used in daily lives. In this research, we use them to generate the perception of new environment and receive the news from communication. This section describes about the adaptation of human perception, which composes of perception from vision and communication to model a crowd simulation.

### 3.1.1 Vision

Vision is one of the most important ability to perception. Since human can learn many things from vision, a crowd simulation should not neglect modeling of vision. This research models the vision of each individual by setting two rules in order to increase realism to our crowd simulation. These rules are limiting vision of each individual when the object or scene is out of sight or when seeing through opaque material. In the technical term, the agents perceive the environmental data by shooting the limited sensor to detect the data as shown in Figure 2.



Figure 2. Vision

### 3.1.2 Communication

This research sets this kind of perception by setting the limited distance to communicate since communication is occurred in finite distance that each person can hear each other. The information can be exchanged immediately when two of individuals reach this limited distance as shown in Figure 3.

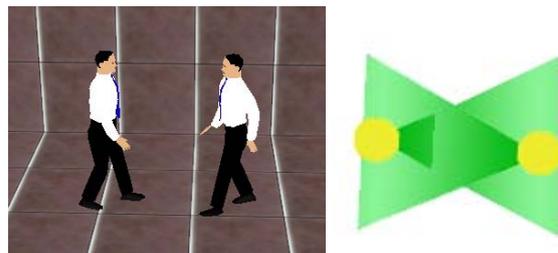


Figure 3. Communication

## 3.2 Cognition

Cognition develops from perception and it can be either large or little amount depending from experiences the one gained from their past experiences of vision and communication. Environment recognition is one of the most used skills in our lives. This research uses this idea to model recognizable crowd in order that each

individual is closer to real human. The recognition of each individual result either from seeing in limited distance as stated in section 3.1.1 or from communicating with other individuals as in 3.1.2. We assign instant recognition of environment to each individual perception. Crowd can remember immediately and its memory is not deleted until the end of the simulation. Moreover, an individual can recognize more environments while moving in the environment. It means that if it can perceive the whole environment then it can also recognize that environment. Knowledge gain from recognition that is used in path construction is called map-knowledge. This map-knowledge is classified into two groups: walkable area and unwalkable area as shown in Figure 4. These data make each individual know the area that can connect to other unknown areas.

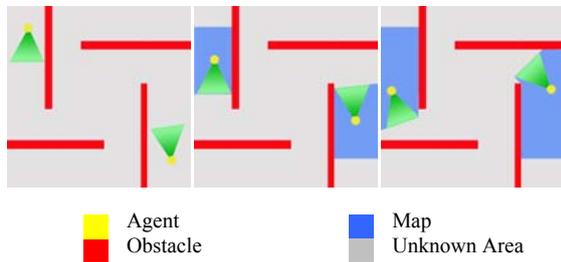


Figure 4. Map construction

### 3.3 Decision

Decision is an important part that affects realism of a behavior model. Path selection is one of the most frequent seen in our daily lives. This section presents the decision model for optimum path selection that yields the most convincing crowd locomotion toward the destination. In the comparison of each possible path can be evaluated from using cost in walking into that path, which composes of the distance, crowd density and the inconvenience of moving. Each actor is weighted by an adjustable parameter and can be written as follows:

$$C = \alpha S_{pi} + \beta \rho_{pi} + \gamma g_{pi}$$

- where  $C$  is the cost in walking into that path  
 $pi$  is the path from present position toward the destination  $i$   
 $S$  is the distance  
 $\rho$  is the crowd density within eyesight  
 $g$  is the inconvenience of moving from the height of the ground at the eye level  
 $\alpha, \beta, \gamma$  is the weight of  $S, \rho, g$  respectively

We use potential field as a method to construct the path of crowd. More details of the algorithm can be found in [Tsa05b]. We start by assign potential at the destination zero value and gradually increase the value when the distance is further. We assume that each individual construct its path from its local knowledge, not calculating from global environment. Each individual can search for its optimum path from using only its knowledge. Since we do not use pre-computed path, we can have two methods of path construction. These are path construction from cover and uncover knowledge, which can be described as follows:

#### 3.3.1 Cover Knowledge

If an individual has global knowledge toward the destination, it means that it know at least one path to the destination. For this reason, we can generate potential field from the destination to the position of that individual directly by constructing from its knowledge-based as described earlier.

#### 3.3.2 Uncover Knowledge

For an individual that has no global knowledge toward the destination, which is called the main destination in this section, it has to observe possible paths that may lead to the main destination. It has to move in the unknown area to observe new environment. The connection area between known an unknown area is assigned to be the minor destination that can lead to the main destination by calculating potential field as described earlier. However, an individual can have more than one connection area, so we evaluate potential field from the connection area of every point by choosing the minimum value of evaluated cost from every connection area of the environment at the end of the time step of the simulation. As a result, an individual can perceive the connection area and the optimum path to choose as shown in Figure 5.

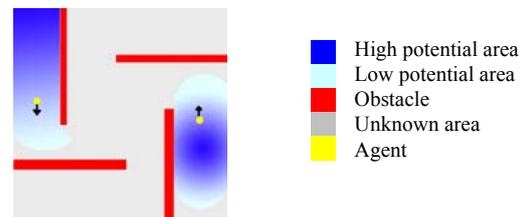


Figure 5. The local potential density map

### 3.4 Locomotion

Crowd locomotion results from path selection in previous process. Normally, a model of an individual motion is only assigning its path toward the destination that does not collide with obstacles and other individuals. However, when it comes to crowd simulation, this is not enough since flocking of massive crowd yield characteristic pattern of

locomotion. Although individuals within the same group share the same destination, it does not mean that each individual has to move toward the destination in the same pattern. Uncertain patterns are occurred since each individual has its own decision that is characterized in the real life crowd. This research models a crowd simulation by using Smoothed Particle Hydrodynamics. Crowd volume can be viewed as particle systems that move in accordance with the rule of fluid dynamics along the direction determined by potential field. It can animate smooth crowd locomotion under various conditions, for example forming a lane when passing narrow area and separating when passing wide area without collision occurred. The rule of smooth crowded locomotion is described in the next section.

### 3.5 Knowledge Sharing

Knowledge sharing is one of common behaviors that can be seen every day. One of these that involves with this research is knowledge sharing from communication as seen in asking for path of tourists. This research models knowledge sharing as describe earlier in section 3.1.2. Knowledge is new environment that it does not know before. Knowledge sharing makes an individual constructing a path to new environment without having seen it earlier by itself. An individual can gain much knowledge if it receives knowledge from other individuals from different areas, and then it can construct the optimized path to the destination faster.

## 4. LOCOMOTION BEHAVIOR

Locomotion behavior is based on Smoothed Particle Hydrodynamics (SPH) which is an interpolation method that approximates the value of a continuous field quantity and its derivative by using discrete sample points. SPH has been widely used for astrophysical problems and fluid dynamics, and has recently gained its popularity among computer graphics researchers. In SPH, the scalar quantity  $A$  at the position  $\mathbf{r}$  is approximated by a summation interpolant.

$$A_s(\vec{r}) = \sum_{j \in N} A_j \frac{m_j}{\rho_j} W(\vec{r}_i - \vec{r}_j, h) \quad (1)$$

Where  $A_j$  is the value of a field quantity at particle  $j$ .  $m_j, \rho_j$  are the mass and density of particle  $j$ .  $N$  is a set of the neighboring particle within the smoothing length,  $h$ .  $W(\vec{r}, h)$  is the weighting function referred as smoothing kernel in SPH. The smoothing length sets  $W=0$  for  $|\vec{r} - \vec{r}_j| > h$  in order to limit the interaction within finite radius. In addition, this smoothing should be even and normalized in order to

correctly represent physical quantities and ensure second order accuracy interpolation. In this work, we used the special-proposed kernels as in Gross et al. [Mul03a] to achieve real time result.

The gradient and laplacian of the summation interpolant can be written as;

$$\nabla A_s(\mathbf{r}) = \sum_{j \in N} m_j \frac{A_j}{\rho_j} \nabla W(\vec{r} - \vec{r}_j, h) \quad (2)$$

$$\nabla^2 A_s(\mathbf{r}) = \sum_{j \in N} m_j \frac{A_j}{\rho_j} \nabla^2 W(\vec{r} - \vec{r}_j, h) \quad (3)$$

Our crowd motion is described by using the equations of fluid flow, which are two equations: Navier-Stokes and continuity. The first equation states that the momentum of the flow is always conserved and any changes in fluid velocity result from self-convection, pressure gradient, internal resistance and the external forces that act upon the fluid. However, the convective term can be omitted in Lagrangian description. Navier-Stokes equation can be written in term of the substantial derivative as follow,

$$\frac{D\vec{v}}{Dt} = -\frac{1}{\rho} \nabla p + \mu \nabla^2 \vec{v} + \vec{f} \quad (4)$$

Where  $\vec{v}, \rho, P, \mu$  are the fluid velocity, density, pressure and the kinematic viscosity, respectively.  $\vec{f}$  is the external body forces in which the gravitational force and user-interaction force are included.

From equation 4, the right hand side implies that there are three forces that responsible for the motion of a particle; pressure force, viscous force and external body force. The sum of these forces is used to calculate the acceleration of particle  $i$

$$\vec{a}_i = \frac{d\vec{v}_i}{dt} = \frac{\sum \vec{f}_i}{\rho_i} \quad (5)$$

$$\sum \vec{f}_i = \vec{f}_i^{pressure} + \vec{f}_i^{visc} + \vec{f}_i^{ext} \quad (6)$$

Where

$$\vec{f}_i^{pressure} = -\sum_{j \in N} m_j \frac{p_i + p_j}{2\rho_j} \nabla W(\vec{r}_i - \vec{r}_j, h) \quad (7),$$

$$\text{and } \vec{f}_i^{visc} = \mu \sum_{j \in N} m_j \frac{\vec{v}_j - \vec{v}_i}{\rho_j} \nabla^2 W(\vec{r}_i - \vec{r}_j, h) \quad (8)$$

Another equation is continuity equation or mass conservation. It states that the rate of density change

of an infinitesimal fluid element equal the total amount of mass per volume entering and leaving the volume occupied by the element. Thus, mass is always conserved during flow.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \bar{v}) = 0 \quad (9)$$

Since the number of particles and their individual mass are constant, the mass conservation can be omitted completely. As a result, we need to evaluate only Navier-Stokes equation to obtain the velocity and the position of each particle.

In our work, we use the Navier-stokes equation to set the rule of our crowd locomotion. The two leftmost terms on the right side of the equation are defined internal interaction within group. These terms are used to achieve smooth movement and handle collision. The external forces which are used to set the direction of crowd are derived in the second stage of the simulation. The amount of force applied is set to be constant, while the direction is resulted from the decision.

## 5. IMPLEMENTATION

Our crowd simulation based on behavior model and SPH which can generate smooth crowd locomotion in interactive time without using pre-computed path. Although it can create a smooth flow of crowd in various situations, our model suits best with the case of crowd escape from unknown environment. The process of crowd simulation is started when the individuals are assigned the random position. The system will repeats the vision creation, map creation, path creation and movement creation processes until the crowd reaches its destination while the animation is shown in Display process as shown in Figure 6. Each of them can be described in details as:

### 5.1 Perception and Recognition

The agent is received knowledge from its perception by shooting the limited sensor to detect the environmental data. The agent of each group can collect the knowledge from individuals in the same group and then merges the knowledge to keep in its map. The map composes of a known environment, connection area and obstacles. The information is then used in decision process to select the path. Their maps can be merged when two individuals have

meeting within the interactive distance as described in section 3.1.2. An agent can also use the new knowledge same as using the knowledge which is gained from its perception.

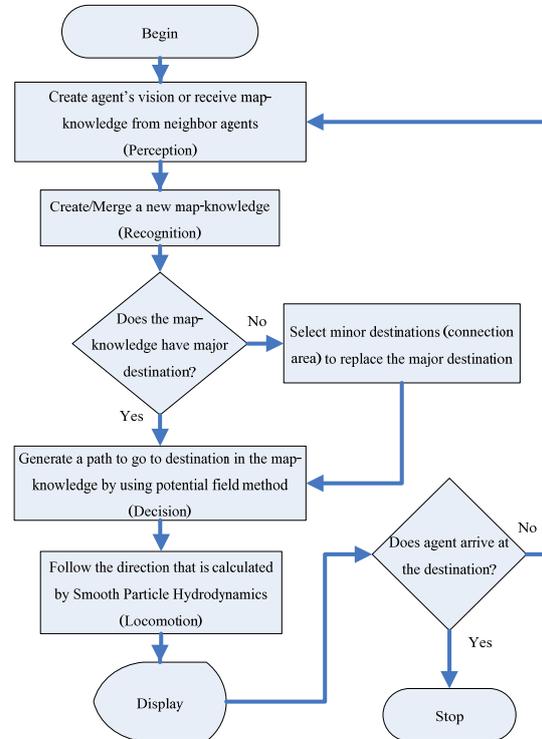


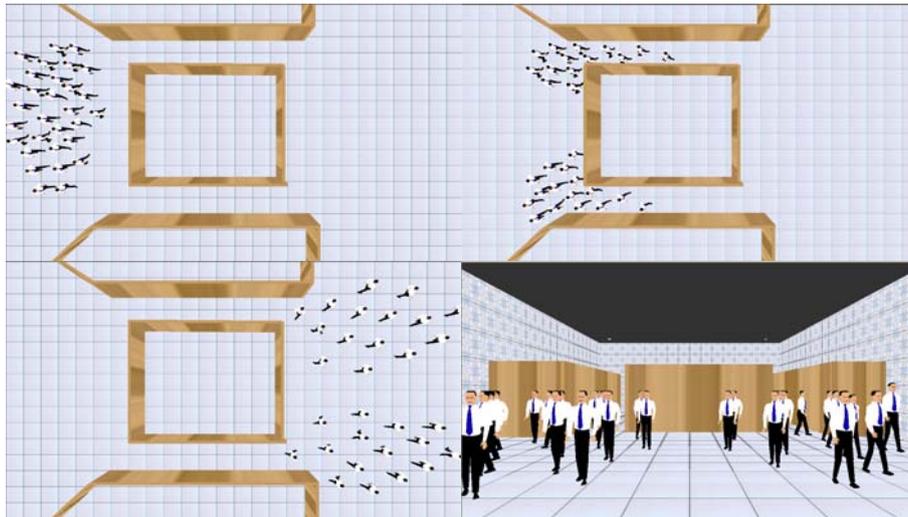
Figure 6. Simulation flow diagram

### 5.2 Decision

Path can be constructed by the agent after creating the map. An agent from each group can select the path toward the destination, using its map. In the case that the map covers the main destination, potential is generated from the destination along the map in the agents' knowledge as described in Section 3.3.1. In another case, potential is generated as described in Section 3.3.2. This generated potential is used to guide the individuals to select the direction, and this selected direction is used as the driving force in locomotion process.

Results	Approaches		
	Proposed Approach	Fluid-based	Agent-based
Non pre-computed global path	✓		
Automatic goal seeking	✓	✓	✓
Making independent decisions	✓		✓
Interactive rate of large crowd simulation	✓	✓	
Natural phenomena of crowd locomotion	✓	✓	

Table 1. Experimental results



**Figure 7. Smooth flow under variety of conditions such as forming lane in narrow area or separating in wide area**

### 5.3 Locomotion

Crowd locomotion can be created after the direction of each group is known. The locomotion can be described by using SPH which is stated in section 4. Crowd volume seems to be collection of fluid particles which move in accordance with the external driving force that act on them. This force is set to have a constant value to keep the walking speed of each individual, while the direction of the force is different since it is based on the potential field approach. When an individual travels to new environment, its knowledge can always be increased.

## 6. EXPERIMENTAL RESULTS

In our experimental results; three types, Agent-based, Fluid-based and Individual-Knowledge-Merge based (IKM), of simulation were compared. The experiments demonstrated how to find a path of the crowd simulation in the unknown complex environment with the same initial conditions. Individuals were archived in random position and tried to the goal.

First, Agent approach based on Autonomous Pedestrians research could fast to seek the goal because it used pre-computed path to guide all individuals. This approach could not generate the loosed way behavior which was seen in the real world. The system could demonstrate independent path selection by self-individual decision but could not demonstrate a natural crowd behavior when the individuals formed a large group.

Next, Fluid approach based on continuum crowds research could seek the goal same as first approach and it could demonstrate a natural large crowd phenomena in real time but could not specify each individual decision behavior.

Finally, IKM approach could seek the goal but did not use pre-computed global path. The individual had a difference direction from its vision to find the destination which consisted of both correct and incorrect path in the beginning. Then, it had the most correct path in the next time when it communicated with each others. Moreover, it could demonstrate natural large crowd phenomena in interactive rate as shown in Figure 7.

The results shown that IKM approach could be used with various situations which were flexible more than pure Agent-based or Fluid-based as shown in Table 1.

## 7. CONCLUSIONS AND FUTURE WORKS

This paper proposes a crowd simulation that coupling fluid dynamics with Agent-based approach, which is used in modeling human decision, path searching and path construction from local knowledge to create a characteristic crowd locomotion. We can animate a crowd simulation that has more visual realism in a path determination and each individual's decision. Our model can simulate smooth crowd locomotion under various environments and conditions. Each individual can recognize the difference knowledge based on their past experiences of perception, it can only generate own appropriate path to find the destination naturally. Moreover, the knowledge of each individual can be merged when they have communication. Individual uses a knowledge sharing to construct a path to new environment without having seen it earlier by itself, because the new knowledge is new environment that it does not know before. An individual can gain much knowledge, if it receives knowledge from other individuals from different areas, and then it can construct the

optimized path to the destination faster. We group the individuals that have the same knowledge and destination for computational time reduction by specifying all individual to use agent's computational results.

In the future work, we can improve the behavioral model for simulate the more realistic human behavior. This paper assumes that the shared knowledge can be used by receiver immediately which reduces a little of realistic recognition behavior of human, because the human must spends the time to learn the knowledge in the real world. We can propose an approach to address this problem by setting the time of perceptions to recognize the environment which is translated to the local knowledge in next time. Moreover, the system can be increased or improved the factors that have influence with decision behavior in several situations.

## REFERENCES

- [Bay04a] Bayazit O., Lien J.-M., Amato N., Better flocking behaviors using rule-based roadmaps. In *Algorithmic Foundations of Robotics V*, Springer Tracts in Advanced Robotics 7, Springer-Verlag Berlin Heidelberg, pp.95–111, 2004
- [Bra03b] Braun A., Bodmann B. E. J., Oliveira L. P. L., and Musse S. R. Modelling individual behavior in crowd simulation. In *Proceedings of Computer Animation and Social Agents 2003*, New Brunswick, USA, IEEE Computer Society. Pp.143-148, 2003
- [Hel00a] Helbing, D.; Farkas, I. and Vicsek, T. "Simulating Dynamical Features of Escape Panic". *Nature*, v. 407, pp.487-490, 2000.
- [Hug03b] HUGHES, R. L., The flow of human crowds. *Annu. Rev. Fluid Mech.* 35, pp.169–182, 2003.
- [Lat04a] Latiff, M. S., and Widyarto S., the crowd simulation for interactive virtual environment., *Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry*, pp.278 – 281, 2004.
- [Mul03a] Müller, M., Charypar, D., and Gross, M. 2003. Particle-based fluid simulation for interactive applications. In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, pp.154–159, 2003.
- [Mur02b] Murakami Y., Minami K., Kawasoe T., and Ishida T., Multi-Agent Simulation for Crisis Management. *KMN, Proceedings of the IEEE Workshop on Knowledge Media Networking*, pp.135 - 139, 2002.
- [Rei95a] REIF J., WANG H.: Social potential fields: A distributed behavioral control for autonomous robots. In K. Goldberg, D. Halperin, J.- C. Latombe, and R. Wilson, editors, *International Workshop on Algorithmic Foundations of Robotics (WAFR)*, A. K. Peters, Wellesley, MA, pp.431–459, 1995.
- [Rey87b] REYNOLDS C.: Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics* 21, 4, pp.25–34, 1987
- [Rey99c] REYNOLDS C.: Steering behaviors for autonomous characters. In *Game Developers Conference*, 1999.
- [Rus94d] RUSSELL S., NORVIG P.: *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1994.
- [San02a] Sanchez G., Latombe J., Using a prm planner to compare centralized and decoupled planning for multi-robot systems, *Conf. on Robotics and Automation*, pp.2112–2119, 2002.
- [Tre06a] Treuille, A. Cooper, S. and Popović, Z., Continuum Crowds. *ACM Transactions on Graphics (TOG)*, pp.1160-1168, 2006.
- [Tsa05b] Tsai, R., Zhao, H., AND Osher, S., Fast sweeping algorithms for a class of hamilton-jacobi equations. *SIAM Journal of Numerical Analysis* 42, 6, 2005.
- [Wei05a] Wei S., and Demetri T., Autonomous pedestrians. *Symposium on Computer Animation*, pp.19 – 28, 2005.

# Biped Cartoon Retrieval Using LBG-Algorithm Based State Vector Quantization

Siriprapa Pakdee

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

siriprapa4901@yahoo.com

Pizzanu Kanongchaiyos

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

pizzanu@cp.eng.chula.ac.th

## ABSTRACT

Efficiency of cartoon motion retrieval depends on indexing method. Keyword based indexing methods for restoring cartoon motion is simple, but sometimes the process of addressing index to the desired motion is not trivial. Although the methods based on clustering clips of motions or keyframes can be used for retrieval automatically, the complexity of time and reliability depend on size of the database. This research proposes a method for biped cartoon motion retrieval using state vectors of keyframes as the motion indexes. The state vectors derived from the input motion is quantized by our enhanced LBG Algorithm to compute the proper number of indexes for the set of input keyframes. The results show that not only the precision and recall of the cartoon motion retrieval is increased accurately but the process for restoring and retrieval is also performed automatically.

## Keywords

Computer Animation, Biped Cartoon, Vector Quantization, State Vector, LBG Algorithm

## 1. INTRODUCTION

Computer games and animation industry at present is growing increasingly. Each animated movie or computer game has a lot of motion information which is expensive for construction and maintenance. In addition, creating a new animated movie or game needs a lot of time and resources, so the reuse of animation components is needed for decreasing the cost and time.

Storing cartoon motions in a database is an important method for reusing cartoon motion. If the storing method of cartoon motion databases is not efficient enough, it will affect the retrieval of cartoon motions. A method for restoring motions using motion index [Keo04a] have proposed a novel technique to speed up similarity search under uniform scaling, based on bounding envelopes for indexing human motions. Liu et al. [Liu05a]

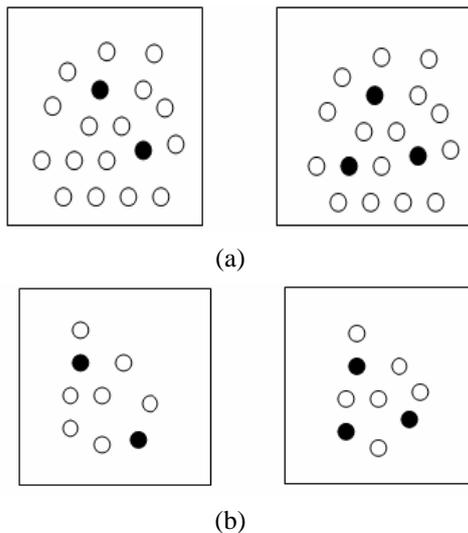
demonstrate a data-driven approach for representing, compressing, and indexing human-motion based on piecewise-linear components obtained using K-means clustering. Forbes and Fiume [For05a] presented a search algorithm to use with sampled motion data. They also developed a representation for motion data introducing a meaningful distance metric for input poses. The methods have been applied to the searching human motion in human motion database for motion synthesizing. Edmunds et al. [Edu05a] demonstrate a method of generating long sequences of motion by performing various similarity-based “joins” on a database of captured motion sequences, while Basu et al. [Bas05a] have proposed how to find similar frames to cluster them. For matching motions, query by example and cluster graph is used to find stored components having same significant DOFs. These methods for storing cartoon motions have some advantage of ability to store the cartoon motion automatically and rapidly, but their time complexity are still not good enough for large database, while their reliabilities of the retrieval motions depend on the size of the database.

This research proposes a method for biped cartoon motion retrieval using state vectors of keyframes as indices. The state vectors derived from the input motion is quantized by our enhanced LBG

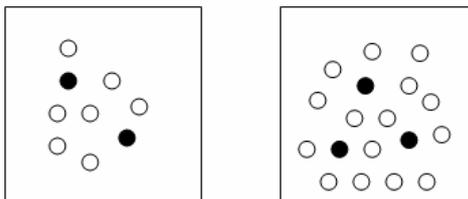
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.  
Copyright UNION Agenc-Science Press, Plzen, Czech Republic.

Algorithm to compute the proper number of indexes for the set of input key frame.

Most algorithms for finding the representation of a vector set properly is based on Linde–Buzo–Gray (LBG) Algorithm [Lin80b, Non05a] and the pair-wise nearest neighbor (PNN) Algorithm [Ari03a] but their efficiency is limited by the user defined number of codewords which is the power of two. Improper size of the codebook causes the density problem such as the set of vector shown in Figure 1, while the proper number of codewords gives more suitable density in set of vector as shown in Figure 2.



**Figure 1. Problem of codebook having improper number of codewords: (a) Code book of two words and three words representing high density vector set (b) Code book size of two words and three words representing vector set having lower density**



**Figure 2. Codebook has a number of codewords corresponding to density of vector.**

Shanbehzadeh and Ogunbona [Sha97a] compared the computational complexity of the pair-wise nearest neighbor (PNN) and Linde–Buzo–Gray (LBG). They showed that for a practical codebook size and training vector sequence, the LBG

algorithm was indeed more computationally efficient than the PNN algorithm.

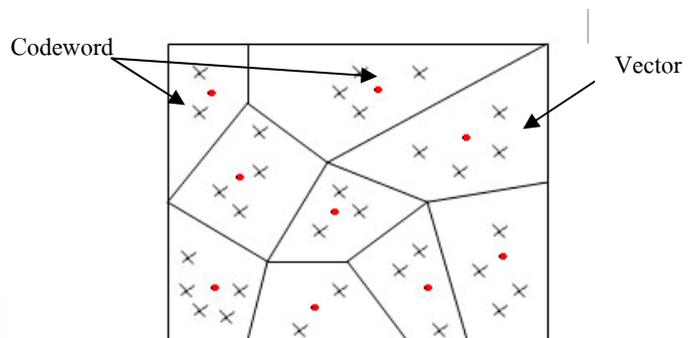
The rest of this paper is organized as follows. In Section 2, we describe the principle of Vector Quantization and proposed how to enhance the LBG algorithm for finding number of codewords automatically in Section 3. State vectors are used to represent cartoon motions in Section 4. Finally, Section 5 shows an experiment of the restoring and retrieval cartoon motion.

## 2. VECTOR QUANTIZATION

The Vector Quantization algorithm is to select  $k$  representatives of vectors from vector space  $R^k$  into a set of vectors. Each  $y_i$  is a codeword, while set of codewords is a codebook. Each codeword  $y_i$  belongs to a nearest neighbor area called Voronoi region defined as follow;

$$V_i = \{x \in R^k : \|x - y_i\| \leq \|x - y_j\|, \text{ for all } j \neq i\}$$

For example, two dimensional vectors in Figure 3 are in their belonging regions. Each group of vectors was represented by their codeword. A codeword must be a vector in the same region as its neighboring vectors.



**Figure 3. An example of vector quantization**

The objective of vector quantization is to represent a group of vectors  $X$  with a codebook  $Y$ . When a vector  $x \in X$  is approximated by a codeword  $y_i$ , an error usually occurs because  $x \neq y_i$ . Mean Square Error can be computed from the sum of distortion error of every group divided by total number of codeword. Efficiency of vector quantization is compared to the distortion error.

Calculation for distance of  $x$  from  $y_i$  starts from defining a distance  $d$  as:

$$d : D^k \times D^k \rightarrow \mathfrak{R}$$

Vector quantization error can be computed using a function  $d(x, y_i)$  which equals to function  $d(x, q(x))$ . In this research, the distance function for finding error is the Euclidean distance;

$$d(x, y) = \sqrt{\sum_{i=1}^k (x - y_i)^2} \quad (1)$$

Therefore the error function is Mean Square Error (MSE)

$$MSE = \frac{1}{N_p} \sum_{p=1}^{N_p} d(x_p, q(x_p)) \quad (2)$$

when  $x_p$  is the  $p^{\text{th}}$  input vector and  $N_p$  is the number of codewords and  $d(x_p, q(x_p))$  is Euclidean distance.

### 2.2 LBG Algorithm

LBG algorithm is a popular method among several codebook design methods. The algorithm is described as follows;

---

**Algorithm 1** Finding quantized representation with LBG algorithm

---

**Pre-condition:**  $N$  is number of codeword,  $T$  is threshold distance

**Post-condition:** All vectors are quantized and represented by randomly choose input vector

**For** all number of codeword **do**

**If**  $d(x, y_i) < d(x, y_j)$  **then**

$x$  is subset of  $y_i$

**else**  $x$  is subset of  $y_j$

**end if**

**end for**

**Calculate distortion**

**While** ( $|D_{prev} - D_{curr}| / D_{curr} > T$ )

**Find** centroid for new codeword

**If**  $d(x, y_i) < d(x, y_j)$  **then**

$x$  is subset of  $y_i$

**else**  $x$  is subset of  $y_j$

**end if**

**end while**

**Final codebook**

---

At the beginning, a vector in set  $X$  will be selected as an initial codebook by randomization and its Voronoi region is defined. By computing the Euclidean distance between each vector and the initial codebook, each vector neighboring to the codeword will be quantized to the same region. Then, the difference between MSE in present and previous iteration is computed to find the stability of the codebook. If stable condition is accepted, the algorithm is terminated. If not, the algorithm will iteratively calculate the centroids vector of each region. The centroid is treated as a new codebook for the initial of next iteration.

### 3. ENHANCEMENT OF LBG ALGORITHM

We develop an enhanced LBG algorithm and use it to find number of representative vectors for each cluster automatically. In this research, the acceptance threshold and acceptance mean square error is user-defined. While comparing MSE from each iteration in the algorithm to the pre-defined threshold and pre-defined MSE that we can insert a new codeword to the codebook. If both values are less than the pre-defined value, the output result is the current codebook. If the MSE is greater than the pre-defined value but the MSE ratio less than threshold, then another codeword is inserted randomly and the LBG algorithm starts again. The enhanced algorithm is shown below and displayed as a diagram in Figure 6.

---

**Algorithm 2** Enhanced LBG algorithm

---

**Pre-condition:**  $S$  is MSE,  $T$  is threshold,  $M=1$

**Post-condition:** All vectors are quantized and represented by randomly choose input vector

**For** all number of codeword **do**

**If**  $d(x, y_i) < d(x, y_j)$  **then**

$x$  is subset of  $y_i$

$x$  is subset of  $y_j$

end if

end for

Calculate distortion

While ( $|D_{prev} - D_{curr}| / D_{curr} > T$ )

Find centroid for new codeword

$M = M + 1$

If  $d(x, y_i) < d(x, y_j)$  then

$x$  is subset of  $y_i$

else  $x$  is subset of  $y_j$

end if

Calculate distortion

end while

while ( $D_{curr} \geq S$ )

Find quantize represent by randomly choose input vector

$M = M + 1$

If  $d(x, y_i) < d(x, y_j)$  then

$x$  is subset of  $y_i$

else  $x$  is subset of  $y_j$

end if

Calculate distortion

end while

Final codebook

#### 4. MOTION REPRESENTATION

In this Section, we will describe how to store cartoon motion into a database. First we represent cartoon motions extracted from keyframes as state vectors which is a set of parameter specifying angular values for each joint of the articulated cartoon character body at is a given time  $t$ . The dimension of the state vector is defined corresponding to the degrees of freedom (DOF) of the cartoon body. For example in Figure 5, the moving arm having three joints have two DOFs, one DOF and one DOF respectively, so each state vector of the arm keyframe in Figure 5 has five dimensions or five elements.

In our example will show storing the motion of an arm which has 3 joints. We use state vectors to represent keyframes of cartoon and use vector

quantization by using enhance LBG algorithm as in figure 4.

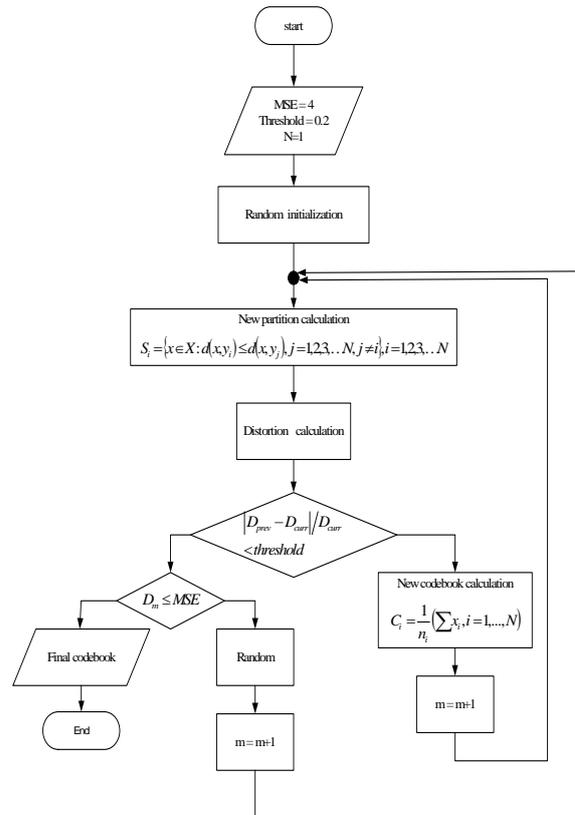


Figure 4. Procedure of enhanced LBG

Moving arm has 10 keyframes which have state vectors as follows:

- q1 ((10, 20), 20, 30), q2 ((10, 20), 20, 35)
- q3 ((10, 20), 25, 35), q4 ((10, 20), 25, 35)
- q5 ((15, 25), 25, 30), q6 ((15, 25), 30, 30)

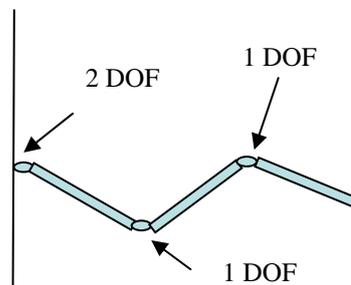


Figure 5. Arm which 1<sup>st</sup> joint has two DOFs (can move two directions). 2<sup>nd</sup> and 3<sup>rd</sup> joint have 1 DOF (can move one direction).

q7 ((15, 25), 30, 25), q8 ((20, 35), 30, 25)  
 q9 ((20, 35), 35, 20), q10 ((20, 35), 35, 15)

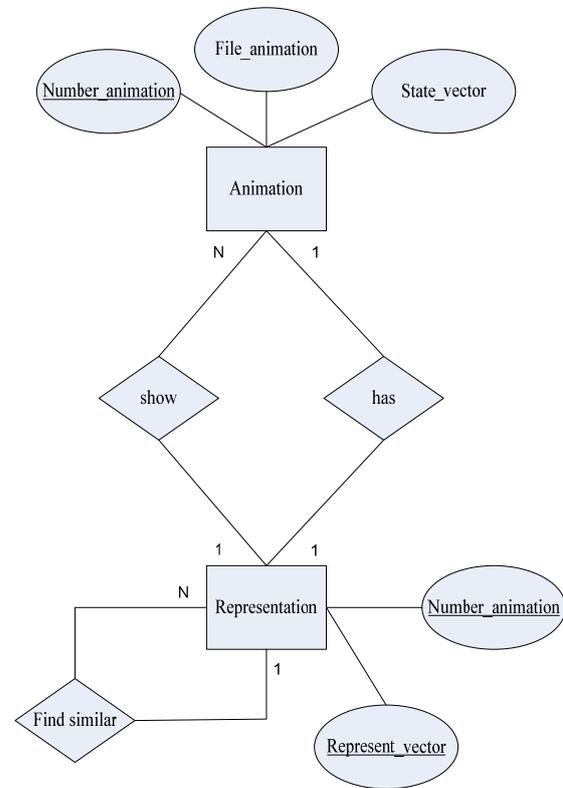
These state vectors are then represented as a codebook using enhance LBG vector quantization algorithm. By repeating this algorithm 3 times, the result of the codebook is shown in table 1. Result is optimal codewords for state vector of our motion as in figure 8 which  $\theta_1$  show state vector which is represent first joint which has two DOFs,  $\theta_2$  show state vector which represent second joint which has one DOF and  $\theta_3$  show state vector which represent third joint which has one DOF.

### 5. EXPERIMENTAL RESULTS

In this research, we set the threshold mentioned in Section 3 to 0.2 and MSE to 4 according to our statistical experimental results. When we got state vectors, we got one initial codeword by random. Define each vector to be same group with the nearest codeword: the nearest distance between codeword and state vector was calculated by Euclidean distance. Calculate Mean Square Error from distortion.

First, we stored skeletal cartoon motion data that was file format MDL1 which MDL1 was Apron Tutorials first 3D skeletal animation format modified by Ronny André Reierstad. The 3D Model 1 format called MDL1 is actually a converted Milkshape 3D ASCII format. You are able to load, import or create your own 3D models and animations with the popular Milkshape 3D editor and export as Milkshape 3D ASCII (\*.txt) files. Skeletal cartoon motion data were stored in skeletal cartoon motion database. Those are the biped cartoon motion: as working, running, jumping, and kicking and are stored into database in state vector form and number that is specified in cartoon through our program. Then we classify information into cluster and find representation of each cluster. We explain and show table that store all cartoon motion in figure 6. In the process, Information of biped cartoon in state vector format and also cartoon motion file and index of cartoon (use sequence of number to represent input cartoon motion by order) is stored into animation table in database by our program. Then we bring query cartoon motion to calculate its state vector which is characteristic of cartoon motion in that time. State vector attribute define motion of each cartoon. Number of state vector is not equal in each cartoon depend on number of keyframes of cartoon motion. State vector also is stored in animation table. So this table uses the attribute File\_animation to represent cartoon motion, Number\_animation

represent cartoon index and State vector represent state vector. When we get state vector of each cartoon motion, we bring its to find representation of state vectors series in each motion by using vector quantization which represent out state vector by Representation table.



**Figure 6. Show E-R model show relation of database that consists of entity “Animation” which has Primary key is Number\_animation and entity “Representation” which primary key is Number\_animation and Representation.**

Representation of each motion different depends on number of keyframes in each cartoon motion. This number of keyframes is flexible by number of motion .Relation of each table can be explained by E-R Model in figure 7 which relation of entity “Animation” and entity “Representation” is 1:1. In finding representation of state vector of motion to store into database and 1: N in getting alike query motion to show and entity “Representation” entity has relation 1: N with itself in finding representation that alike as query motion.

The retrieval of biped cartoon motion is tested by using state vector instead of keyframe of cartoon. Then using vector quantization by LBG algorithm. We tested by storing 30 motions into database.

Database had several biped cartoon motions which were synthetic data.

In retrieval of cartoon motions, we retrieve from motion of cartoon which is stored 30 examples. We get motion information from desired cartoon. In this case we use walking to search motion information which is the same or alike as motion information in database as in figure 7. By quantizing vector of query cartoon, the result is a codebook, and then we bring it to compare with each codebook in database. We compare the different distance of each codebook in database with the motion we need, using Euclidean distance. We show these values in table 1 and these values must not exceed 15 .We show result of desired motion in figure 8.

The motion retrievals	The distance of each codebook.
Motion 1	12.41
Motion 2	10.52
Motion 3	8.60
Motion 4	13.35
Motion 5	11.30

Table 1. Show value of the different distance of each codebook in database with the motion we need.

Then we compare output with input. We calculated the recall that was number of items retrieved and relevant and total relevant in motion database ratio and calculated the precision that was number of items retrieved and relevant and total retrieve ratio. We repeated this 100 time and consider the entire of the precision and recall which results are satisfactory.

## 6. CONCLUSION

In this paper we have proposed a method for biped cartoon motion retrieval using state vectors of keyframes as indexes. Input motion give state vector is quantized by enhanced LBG Algorithm, developed by our, to calculate indexes for set of input key frame. Because of the motion of cartoons which have difference the number of keyframes, are stored in database. If we define the number of codewords are constant, the result of codewords are calculated is not flexible to number of frame. The result is the retrieval of cartoon motion that is more accurately and automatically.

In the future works, we will try to speed up the computation of the state vector and the codebooks

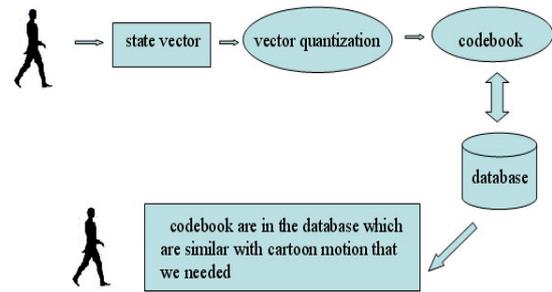


Figure 7. Show procedure of the retrieval.

If we use sorting techniques and searching techniques to sort and search the motion of cartoon, the response time to find the result of the retrievals would be decrease. We compare the response time of the retrieval that use our method, between differences searching technical as Rtree search algorithm and FastScan algorithm. The accuracy of retrieval by our method, depend on searching techniques by using same database. If the sizes of database are increase, the result could vary.

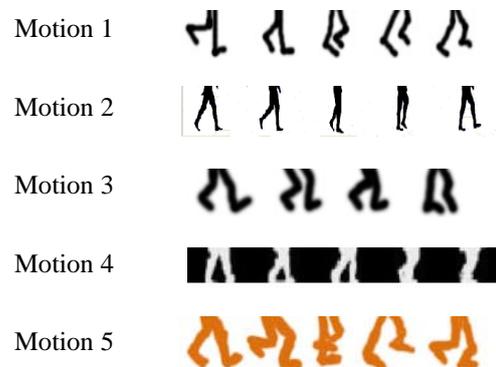


Figure 8. Show example output of walking motion.

## 7. REFERNCES

- [Ari03a] O. Arikan, David A., F. James and F. O'Brien. Motion Synthesis from Annotations. ACM Transactions on Graphics, pp. 402 – 408, 2003.
- [Bas05a] S. Basu, S. Shanbhag and S. chandran. Search and Transitioning for Motion Captured Sequences. Proceedings of the ACM symposium on Virtual reality software and technology VRST '05, pp. 220–223, 2005.

- [Edu05a] T. Edmunds, S. Muthukrishnan, S. Sadhukhan and S. Sueda. MoDB: Database System for Synthesizing Human Motion. Proceedings of the 21st International Conference on Data Engineering (ICDE'05), pp. 1131–1132, 2005.
- [For05a] K. Forbes and E. Fiume. An Efficient Search Algorithm for Motion Data Using Weighted PCA. Proceedings of the 2005 ACM SIGGRAPH / Eurographics symposium on Computer animation, pp. 67 – 76, 2005.
- [Keo04a] E. J. Keogh, T. Palpanas, V. B. Zordan, D. Gunopulos, and M. Cardle. Indexing large human-motion databases. In VLDB, pp. 780–791, 2004.
- [Liu05a] G. Liu, J. Zhang, W. Wang, and L. McMillan. A System for Analyzing and Indexing Human-Motion Databases. In SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data, pp. 924–926, 2005.
- [Lin80b] Y. Linde, A. Buzo and R. Gray. An Algorithm for Vector Quantizer Design. IEEE Transactions on [legacy, pre - 1988], Vol. 28, pp. 84-95, 1980.
- [Non05a] A. Nongnuch, A. Surarerks. An Algorithm for Vector Quantization Using Density Estimation. Master's thesis, Faculty of Engineering, Chulalongkorn University, 2005.
- [Pat01a] G. Patanè. Unsupervised Learning on Tradition and Distributed Systems. Master's thesis, University of Palermo, 2001.
- [Sha97a] J. Shanbehzadeh and P. O. Ogunbona. On the Computational Complexity of the LBG and PNN Algorithms. IEEE Transactions on Image processing, Vol. 6, No. 4, pp. 614-616, 1997.
- [Gra84a] R. Gray. Vector Quantization, IEEE ASSP Magazine, pp. 4-29, 1984.

