

**The 19<sup>th</sup> International Conference in Central Europe on Computer  
Graphics, Visualization and Computer Vision**

in co-operation with

**EUROGRAPHICS**

# **W S C G ' 2011**

## **Poster Papers Proceedings**

University of West Bohemia  
Plzen  
Czech Republic

January 31 – February 3, 2011

*Co-Chairs*

**Gladimir Baranoski, University of Waterloo, Canada**  
**Vaclav Skala, University of West Bohemia, Czech Republic**

*Edited by*  
Vaclav Skala

***WSCG'2011 Poster Papers Proceedings***

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

Managing Editor: Vaclav Skala

Published and printed by:  
Vaclav Skala – Union Agency  
Na Mazinách 9  
CZ 322 00 Plzen  
Czech Republic

Hardcopy: *ISBN 978-80-86943-81-7*

# WSCG 2011

## International Program Committee

Balcisoy, S. (Turkey)  
Baranoski, G. (Canada)  
Benes, B. (United States)  
Benoit, C. (France)  
Bilbao, J. (Spain)  
Biri, V. (France)  
Bittner, J. (Czech Republic)  
Bouatouch, K. (France)  
Buehler, K. (Austria)  
Coquillart, S. (France)  
Daniel, M. (France)  
de Geus, K. (Brazil)  
Debelov, V. (Russia)  
Feito, F. (Spain)  
Ferguson, S. (United Kingdom)  
Flaquer, J. (Spain)  
Gallo, G. (Italy)  
Gavrilova, M. (Canada)  
Gudukbay, U. (Turkey)  
Gutierrez, D. (Spain)  
Havemann, S. (Austria)  
Havran, V. (Czech Republic)  
Chmielewski, L. (Poland)  
Chover, M. (Spain)  
Jansen, F. (Netherlands)  
Klosowski, J. (United States)  
Lee, T. (Taiwan)  
Max, N. (United States)  
Molla Vaya, R. (Spain)  
Murtagh, F. (Ireland)  
Pasko, A. (United Kingdom)  
Pedrini, H. (Brazil)  
Platis, N. (Greece)  
Puppo, E. (Italy)  
Purgathofer, W. (Austria)  
Rojas-Sola, J. (Spain)  
Rokita, P. (Poland)  
Rosenhahn, B. (Germany)  
Rudomin, I. (Mexico)  
Sakas, G. (Germany)  
Segura, R. (Spain)  
Schumann, H. (Germany)  
Skala, V. (Czech Republic)  
Slavik, P. (Czech Republic)  
Sochor, J. (Czech Republic)  
Stroud, I. (Switzerland)  
Teschner, M. (Germany)  
Theoharis, T. (Greece)  
Tokuta, A. (United States)  
Vergeest, J. (Netherlands)  
Wu, S. (Brazil)  
Wuethrich, C. (Germany)  
Zara, J. (Czech Republic)  
Zemcik, P. (Czech Republic)  
Zitova, B. (Czech Republic)

WSCG 2011 was supported by



SILICON GRAPHICS s.r.o



**Microsoft®**

Microsoft, s.r.o. ČR



Faculty of Applied Sciences

Dept. of Computer Science &  
Engineering

# WSCG 2011

## Board of Reviewers

Akleman, E. (United States)  
Ariu, D. (Italy)  
Assarsson, U. (Sweden)  
Aveneau, L. (France)  
Balcisoy, S. (Turkey)  
Battiato, S. (Italy)  
Benes, B. (United States)  
Benoit, C. (France)  
Biasotti, S. (Italy)  
Bilbao, J. (Spain)  
Biri, V. (France)  
Bittner, J. (Czech Republic)  
Bosch, C. (France)  
Bouatouch, K. (France)  
Boukaz, S. (France)  
Bouville, C. (France)  
Bruni, V. (Italy)  
Buehler, K. (Austria)  
Cakmak, H. (Germany)  
Camahort, E. (Spain)  
Capek, M. (Czech Republic)  
CarmenJuan-Lizandra, M. (Spain)  
Casciola, G. (Italy)  
Coquillart, S. (France)  
Correa, C. (United States)  
Cosker, D. (United Kingdom)  
Daniel, M. (France)  
de Amicis, r. (Italy)  
de Geus, K. (Brazil)  
Debelov, V. (Russia)  
Domonkos, B. (Hungary)  
Drechsler, K. (Germany)  
Duke, D. (United Kingdom)  
Dupont, F. (France)  
Durikovic, R. (Slovakia)  
Eisemann, M. (Germany)  
Erbacher, R. (United States)  
Erleben, K. (Denmark)  
Farrugia, J. (France)  
Feito, F. (Spain)  
Ferguson, S. (United Kingdom)  
Fernandes, A. (Portugal)  
Flaquer, J. (Spain)  
Fontana, M. (Italy)  
Fuenfzig, C. (France)  
Gallo, G. (Italy)  
Galo, M. (Brazil)  
Garcia Hernandez, R. (Spain)  
Garcia-Alonso, A. (Spain)  
Gavrilova, M. (Canada)  
Giannini, F. (Italy)  
Gonzalez, P. (Spain)  
Grau, S. (Spain)  
Gudukbay, U. (Turkey)  
Guggeri, F. (Italy)  
Gutierrez, D. (Spain)  
Habel, R. (Austria)  
Hall, P. (United Kingdom)  
Hansford, D. (United States)  
Haro, A. (United States)  
Hasler, N. (New Zealand)  
Havemann, S. (Austria)  
Havran, V. (Czech Republic)  
Hernandez, B. (Mexico)  
Herout, A. (Czech Republic)  
Horain, P. (France)  
House, D. (United States)  
Chaine, R. (France)

Chaudhuri, D. (India)  
Chmielewski, L. (Poland)  
Chover, M. (Spain)  
Iwasaki, K. (Japan)  
Jansen, F. (Netherlands)  
Jeschke, S. (Austria)  
Jones, M. (United Kingdom)  
Jones, M. (United States)  
Juettler, B. (Austria)  
Kheddar, A. (Japan)  
Kim, H. (Korea)  
Klosowski, J. (United States)  
Kohout, J. (Czech Republic)  
Kurillo, G. (United States)  
Kyratzi, S. (Greece)  
Lanquetin, S. (France)  
Lay Herrera, T. (Germany)  
Lee, T. (Taiwan)  
Lee, S. (Korea)  
Leitao, M. (Portugal)  
Liu, D. (Taiwan)  
Liu, S. (China)  
Lutteroth, C. (New Zealand)  
Madeiras Pereira, J. (Portugal)  
Maierhofer, S. (Austria)  
Manzke, M. (Ireland)  
Marras, S. (Italy)  
Maslov, O. (Russia)  
Matey, L. (Spain)  
Matkovic, K. (Austria)  
Max, N. (United States)  
Meng, W. (China)  
Mestre, D. (France)  
Michoud, B. (France)  
Mokhtari, M. (Canada)  
Molla Vaya, R. (Spain)  
Montrucchio, B. (Italy)  
Muehler, K. (Germany)  
Murtagh, F. (Ireland)  
Nishio, K. (Japan)  
OliveiraJunior, P. (Brazil)  
Oyarzun Laura, C. (Germany)  
Pan, R. (China)  
Papaioannou, G. (Greece)  
Pasko, A. (United Kingdom)  
Pasko, G. (Cyprus)  
Patane, G. (Italy)  
Patow, G. (Spain)  
Pedrini, H. (Brazil)  
Peters, J. (United States)  
Pina, J. (Spain)  
Platis, N. (Greece)  
Puig, A. (Spain)  
Puppo, E. (Italy)  
Purgathofer, W. (Austria)  
Reshetov, A. (United States)  
Richardson, J. (United States)  
Richir, S. (France)  
Rojas-Sola, J. (Spain)  
Rokita, P. (Poland)  
Rosenhahn, B. (Germany)  
Rudomin, I. (Mexico)  
Sakas, G. (Germany)  
Salvetti, O. (Italy)  
Sanna, A. (Italy)  
Segura, R. (Spain)  
Sellent, A. (Germany)  
Shesh, A. (United States)  
Schultz, T. (United States)  
Schumann, H. (Germany)  
Sirakov, N. (United States)  
Skala, V. (Czech Republic)  
Slavik, P. (Czech Republic)  
Sochor, J. (Czech Republic)  
Sousa, A. (Portugal)  
Srubar, S. (Czech Republic)  
Stroud, I. (Switzerland)  
Subsol, G. (France)  
Sundstedt, V. (Sweden)  
Tang, M. (China)  
Tavares, J. (Portugal)  
Teschner, M. (Germany)  
Theoharis, T. (Greece)  
Theussl, T. (Saudi Arabia)  
Tokuta, A. (United States)  
Tomori, Z. (Slovakia)  
Torrens, F. (Spain)  
Trapp, M. (Germany)

Umlauf, G. (Germany)  
Vazques, P. ()  
Vergeest, J. (Netherlands)  
Vitulano, D. (Italy)  
Vosinakis, S. (Greece)  
Walczak, K. (Poland)  
Weber, A. (Germany)  
Wu, S. (Brazil)  
Wuensche, B. (New Zealand)  
Wuethrich, C. (Germany)

Yoshizawa, S. (Japan)  
Yue, Y. (Japan)  
Zara, J. (Czech Republic)  
Zemcik, P. (Czech Republic)  
Zhu, Y. (United States)  
Zhu, J. (United States)  
Zitova, B. (Czech Republic)



# WSCG 2011

## Poster Papers Proceedings

### Contents

- Santos,R.M.G., De Martino,J.M., Mesquita,E.: Computer Simulation and Visualization of Mechanical Wave Propagation Phenomena in Continuous Medium 1
- Matyunin, S., Vatolin, D., Smirnov, M.: Fast Temporal Filtering of Depth Maps 5
- Sisojevs,A., Glazs,A.: An Efficient Approach to Direct NURBS Surface Rendering for Ray Tracing 9
- Srubar,S., Surkala,M.: Comparison of Mean Shift Algorithms and Evaluation of Their Stability 13
- Kobori,K., Maekawa,N., Nishio,K.: Generation of Indian Ink Painting Image from Two-Dimensional Image Data 17
- Duplex,B., Gesquire,G., Daniel,M., Perdu,F.: Deformation Exchange between Adjacent Physical Code Geometries 21
- Marek,J., Uhrinova,I., Demjen,E., Tomori,Z.: Parametric 3D model of sperm cell surface applied in confocal microscopy 25
- Martinez,A., Jimenez,J., Paulano,F., Pulido,R., Feito,F.: An exact hierarchical geometric model. Combining remeshing and spatial decomposition 29
- Nishio,K., Hiraoka,K., Yamashita,C., Kobori,K.: Construction of Panoramic Depth Image 33
- Behal,D., Dadova,J., Uhlíkova,I.: 3D extension of web 37
- Florek,M., Hauser,H.: Interactive Bivariate Mode Tree 41
- Cassidy,P., Kilburn,T., Salemink,V., Bailey,R., Bischof,H.: Improving the Visualization of Galactic Events Using Pixar's RenderMan 45
- Kefi,M., Richard,P., Barichard,V.: Use Virtual Reality and Constrained Programming Techniques in Interactive 3D Objects Layout 49



# Computer Simulation and Visualization of Mechanical Wave Propagation Phenomena in Continuous Medium

Rodrigo Mologni G. dos Santos

DCA, FEEC, Unicamp  
Av. Albert Einstein, 400  
Cidade Universitária  
13083-852, Campinas, SP, Brazil

mologni@dca.fee.unicamp.br

José Mario De Martino

DCA, FEEC, Unicamp  
Av. Albert Einstein, 400  
Cidade Universitária  
13083-852, Campinas, SP, Brazil

martino@dca.fee.unicamp.br

Euclides de Mesquita Neto

DMC, FEM, Unicamp  
R. Mendeleev, 200  
Cidade Universitária  
13083-860, Campinas, SP, Brazil

euclides@fem.unicamp.br

## ABSTRACT

This paper presents a simulation software for mechanical wave propagation phenomena and a visualization environment developed to help the analysis of the numerically synthesized dynamic solutions of boundary value problems of solid mechanics. The simulation software is used by researchers and engineers to study and gain insights into complex problems related to the mechanics of continuous media. The large amount of data generated by the numerical solution of wave propagation problems makes the interpretation of the results a hard daunting task without the support of techniques to graphically display these data. The visualization environment that was developed supports different visualization strategies of a variety of parameters and it has been used to improve the understanding of the simulation results and to validate the numerical simulator as well as the underlying mathematical-physical modeling.

## Keywords

Computer simulation, scientific visualization, mechanical waves.

## 1. INTRODUCTION

The study of the mechanical wave propagation phenomena and their consequences usually requires computational methods for the resolution of complex problems in mechanics of continuous media, such as finite and boundary element methods. These computational methods consist of mathematical modeling and computer simulation of physical problems, which replace in many cases the use of costly prototypes and laboratory setups. However, the computer simulation produces a large amount of numerical data that makes the procedure of the analysis of the results an arduous and extremely difficult task without the support of data visualization techniques. With the support of visualization, researchers and engineers can more quickly validate the processes of mathematical modeling and computer simulation of the physical problems; and

also they can more easily understand the results generated in the process of computer simulation, which allows a more efficiently study of the physical problems.

## 2. RELATED WORKS

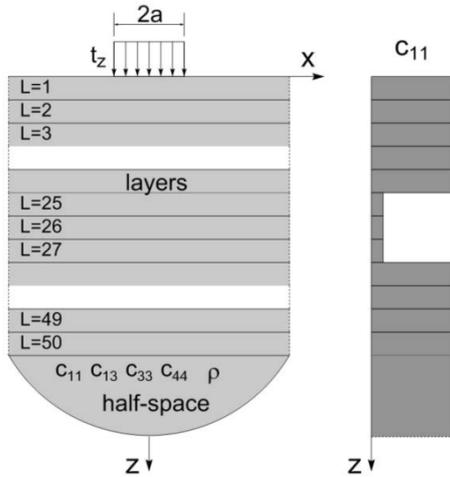
The present work is concentrated on wave propagation in solid continua, particularly in unbounded domains. The problems to be solved are very complex and solution can only be accomplished by resorting to numerical methods. The numerical solution for two-dimensional wave problems can be found, exemplarily, in the works of Rajapakse and Wang [Raj93] and Barros et al. [Bar99]. More recently three-dimensional problems have been solved by Mesquita and his co-workers [Mes09].

## 3. MATHEMATICAL MODELING AND COMPUTER SIMULATION

To illustrate the features of the simulation software a complex wave propagation problem will be numerically solved. The problem solution, that is, the displacement components of the wave field solution will be analyzed and interpreted with the aid of the visualization environment. The problem at hand can be depicted at Fig. 1. It represents a series of 50 horizontal transversely isotropic (visco)elastic layers excited harmonically by a surface stress distribution

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.

$t_z(x, z = 0)$ . The problem has a stationary character and the analysis is performed in the frequency domain. The layers are supported by a transversely isotropic half-space.



**Figure 1. The geomechanical problems: horizontal layers on a half-space**

Every layer is governed by a set of differential equations, which can be expressed in terms of the horizontal  $u_x$  and vertical  $u_z$  displacement components as:

$$\begin{aligned} c_{11}u_{x \rightarrow xx} + c_{44}u_{x \rightarrow zz} + (c_{13} + c_{44})u_{z \rightarrow xz} + \rho\omega^2 u_x &= 0 \\ c_{44}u_{z \rightarrow xx} + c_{33}u_{z \rightarrow zz} + (c_{13} + c_{44})u_{x \rightarrow xz} + \rho\omega^2 u_z &= 0 \end{aligned} \quad (1)$$

In Eq. 1,  $c_{ij}$  ( $ij = 11, 13, 33, 44$ ) and  $\rho$  are constitutive parameters of the elastic layer [Bar99]. The circular frequency is designated by  $\omega$ . The prescribed boundary conditions are the stress free surface outside the loaded area ( $-a < x < a, z = 0$ ) and the Sommerfeld radiation condition at the layers and at the underlying half-space.

As can be seen in Fig. 1, the constitutive parameters of all the layers and of the half-space are the same, except for layers 25 to 27, in which there is an abrupt change in this parameters. Fig. 1 also shows the variation of the constitutive parameter  $c_{11}$  throughout the layers. This parameter discontinuity generates a strong impedance change with significant influence on the wave propagation pattern, as will be discussed later on this article.

A typical numerical expression for the harmonic wave displacement solution is given by [Bar99]:

$$\begin{aligned} u_x(x, z) &= \frac{\delta}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \sum_{m=1}^2 (\bar{\omega}(\xi_m) A_m e^{-\delta \xi_m z} - \bar{\omega}(\xi_m) B_m e^{\delta \xi_m z}) e^{i\delta \xi_m x} d\xi \\ u_z(x, z) &= \frac{\delta}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \sum_{m=1}^2 (A_m e^{-\delta \xi_m z} + B_m e^{\delta \xi_m z}) e^{i\delta \xi_m x} d\xi \end{aligned} \quad (2)$$

## 4. COMPUTER VISUALIZATION

To help the analysis and understanding of the simulation results, the simulated data are enriched with additional application domain specific information.

### Displacement and relative position

Propagation of mechanical waves causes the particles of matter to oscillate around their resting position. In simulation results, these oscillations are generated by action of the displacement components. Thus the displacement of an oscillating particle from its equilibrium position at time instant ( $t$ ) is given by planar vector:

$$\vec{d}_t = (\vec{d}_{x,t}, \vec{d}_{z,t}) \Rightarrow \begin{cases} \vec{d}_{x,t} = \text{Re}(u_x) \cos t - \text{Im}(u_x) \sin t \\ \vec{d}_{z,t} = \text{Re}(u_z) \cos t - \text{Im}(u_z) \sin t \end{cases} \quad (3)$$

Where Re and Im stand for the real and imaginary parts of a complex number, respectively.

The relative position of an oscillating particle, i.e., its new position after displacement, is given by  $P_t = P + \vec{d}_t = (x_t, z_t)$ . Where  $P = (x, z)$  is the resting or equilibrium position and  $\{t \in \mathbb{R} \mid 0 \leq t < 2\pi\}$ .

### Displacement trajectory

When the oscillating particle movement is composed by overlapping simple harmonic motions of same frequency in both coordinate axes, then its oscillation describes an elliptical trajectory. As an ellipse can be drawn from its vertices, then the displacement trajectory of an oscillating particle can be represented by an ellipse whose vertices are the relative positions  $P_{t_k}$  to  $k = \{1, 2, 3, 4\}$ , where  $t_k$  is:

$$t_k = \frac{\tan^{-1} \left\{ \frac{2[\text{Re}(u_x) \text{Im}(u_x) + \text{Re}(u_z) \text{Im}(u_z)]}{\text{Im}(u_x)^2 - \text{Re}(u_x)^2 + \text{Im}(u_z)^2 - \text{Re}(u_z)^2} \right\} + (k-1)\pi}{2} \quad (4)$$

### Aspect ratio

Since the displacement trajectory is elliptical, then the characteristics of an ellipse can also be used to enrich the simulation results. One is the aspect ratio, that is, the ratio ( $b/a$ ) of the semi-major ( $a$ ) and semi-minor ( $b$ ) axes of an ellipse, where:

$$a = \begin{cases} \|\vec{d}_{t_1}\|, \|\vec{d}_{t_1}\| > \|\vec{d}_{t_2}\| \\ \|\vec{d}_{t_2}\|, \|\vec{d}_{t_1}\| < \|\vec{d}_{t_2}\| \end{cases}, \quad b = \begin{cases} \|\vec{d}_{t_1}\|, \|\vec{d}_{t_1}\| < \|\vec{d}_{t_2}\| \\ \|\vec{d}_{t_2}\|, \|\vec{d}_{t_1}\| > \|\vec{d}_{t_2}\| \end{cases} \quad (5)$$

### Rotational movement sense

The rotational movement sense ( $R$ ) of the elliptical trajectory of oscillating particles can be obtained by vector dot product between the normal vectors to the planes of the trajectories ( $\vec{d}_{t_1} \times \vec{d}_{t_2}$ ) and the medium

( $\vec{n}$ ); see Eq. 6. Thus, if result is negative, then  $R$  is clockwise (CW); otherwise, if result is positive, then  $R$  is counter-clockwise (CCW).

$$R = \begin{cases} CW, & \vec{d}_1 \times \vec{d}_2 \cdot \vec{n} < 0 \\ CCW, & \vec{d}_1 \times \vec{d}_2 \cdot \vec{n} > 0 \end{cases} \quad (6)$$

### Phase angle

The phase angle of the waves on each oscillating particle in the  $z$ -axis coordinate is given by:

$$\phi_z = \tan^{-1} \left[ \frac{\text{Im}(u_z)}{\text{Re}(u_z)} \right], \quad -\pi \leq \phi_z \leq \pi \quad (7)$$

## 5. RESULTS AND DISCUSSION

To illustrate the power and usefulness of the visualization strategy adopted, the numerical solution of the problem described in Section 3 is used as a case study. The problem is a two-dimensional, plain strain solution [Lai09], determined at the  $(x, z)$  plane. For the calculated solution the parameters are:  $a = 1m$ ,  $\omega = 1 \text{ rad/s}$ ,  $\rho = 1 \text{ kg/m}^3$ ,  $t_z = 1N$ ,  $c_{11} = 6 \text{ N/m}^2$ ,  $c_{13} = 4 \text{ N/m}^2$ ,  $c_{33} = 6 \text{ N/m}^2$ , and  $c_{44} = 1 \text{ N/m}^2$  for all the layers and the half-space, except for layers 25 to 27 in which  $c_{11}$ ,  $c_{13}$ ,  $c_{33}$ , and  $c_{44}$  are reduced to 10%. The data are generated on a uniform rectilinear grid composed of 5,151 (101 x 51) sampling points.

### Visualization of the displacement

Fig. 2 depicts the vertical component of the displacement field  $(x, z, u_z)$  within the solution range of the problem. The location  $(-a < x < a, z = 0)$  of the applied vertical excitation  $t_z$  and the "fault", that is the discontinuity in the constitutive parameters of layers 25 to 27, can be clearly recognized. Another issue that can be observed in these components is that there is no wave reflection at the end of calculation domains. On the other hand, the impedance discontinuity at layers 25 to 27 clearly causes wave transmission and reflection, distorting the wave pattern relative to the homogeneous domain.

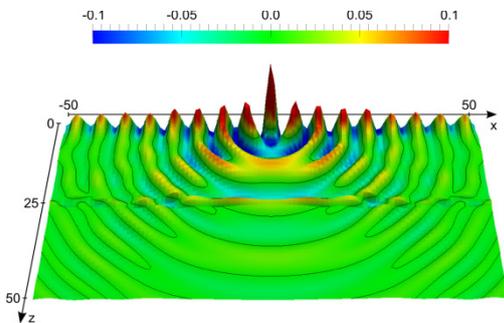


Figure 2.  $u_z$  vertical component at time  $0.19 \cdot 2\pi$

### Visualization of the aspect ratio

In homogeneous solids there are two broad classes of linear waves. The body waves and the surface or interface waves [Gra91]. Body waves such as the dilatational and shear waves present a rather rectilinear trajectory and they tend to cross a 2D homogeneous medium in a circular pattern, starting from the perturbation source. Surface waves such as the Rayleigh waves tend to have retrograde elliptical trajectories and are limited to the surroundings of surfaces. Fig. 3 shows an overview of the particle trajectory aspect ratio. Depending on the constitutive parameters of the medium, the values for the aspect ratio around 0.6 to 0.7 indicate the presence of Rayleigh waves [Gra91]. In Fig. 3, the elliptical Rayleigh wave trajectories are characterized by the color range going from green to yellow, as indicated in the scale (see black isolines).

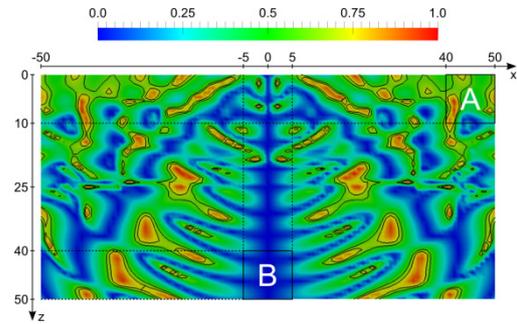
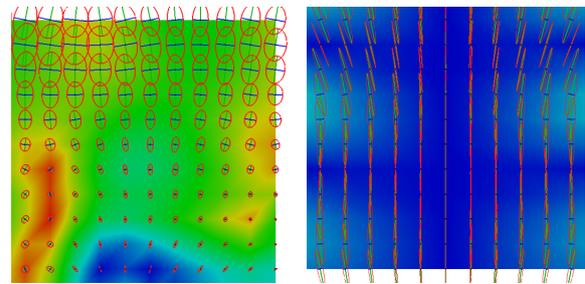


Figure 3. Aspect ratio of the elliptical trajectories

### Visualization of the displacement trajectory

At the free surface  $(x, z = 0)$ , as the analysis moves away from the energy source, the trajectories became elliptical and decrease very rapidly, as can be seen in Fig. 4a, depicting the trajectories within the range  $(40 \leq x \leq 50, 0 \leq z \leq 10)$ . These rapidly decaying elliptical trajectories indicate the presence of Rayleigh waves. The trajectories at the range  $(-5 \leq x \leq 5, 40 \leq z \leq 50)$  are shown in Fig. 4b. The trajectories show a predominance of body waves, characterized by the almost-rectilinear character of the displacements. As the distance from the surfaces or interfaces increases, the body forces tend to dominate the propagation pattern.



(a) Region A in Fig. 3 (b) Region B in Fig. 3

Figure 4. Displacement trajectories

## Visualization of the rotational movement sense

The Rayleigh surface waves present a retrograde elliptical trajectory, as already described. Fig. 5 indicates the rotation sense of the particle trajectories. It can be clockwise (blue) or counter-clockwise (red). Now the trajectories along near the surface ( $z = 0$ ) in the positive  $x$ -direction ( $x > a$ ) are considered. Joining the information of Fig. 3, which states that aspect ratio of the elliptical trajectory near the surface has a value around 0.65, with the information of Fig. 4a, showing the elliptical character of the trajectory, and information of Fig. 5, stating the particle rotation sense is counter-clockwise, it may be established with a fair amount of confidence that the waves propagating at this region are predominantly Rayleigh waves.

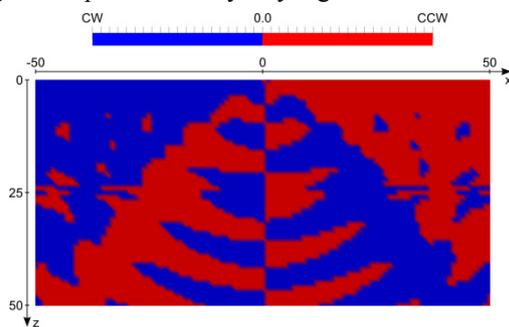


Figure 5. Rotational movement sense of the trajectories

## Visualization of the phase angle

An analysis of Fig. 6, the vertical component, shows clearly two types of wave fronts. Near the surface ( $z = 0$ ) and moving towards the edge of the domain, the wave front end to be plane. Plane wave fronts are typical of Rayleigh wave propagating in two dimensions. This picture helps corroborating that Rayleigh waves are the predominant propagating wave type near the surface and away from the energy source. The second type of wave front present an almost circular character and is typical of body waves, that is, dilatational and shear waves propagating in two dimensions [Gra91].

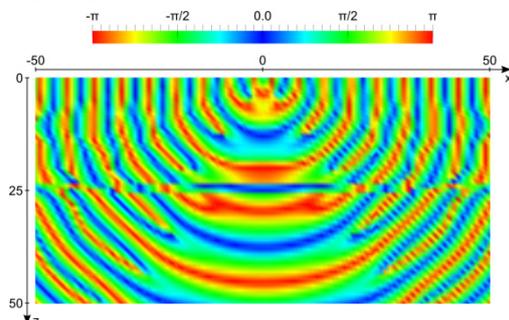


Figure 6. Phase angle of the  $u_z$  vertical component

## 6. CONCLUSIONS

The developed simulation software has been applied to help analyzing the wave propagation pattern resulting from the numerical solution of a very complex geomechanical problem. The numerical tools implemented in the simulation software, together with previous knowledge on the properties of wave types in solids, allowed the problem analyst to understand the various wave types that propagated through the medium and also to determine the region of their respective predominance. With the visualization environment developed, researchers can now concentrate their efforts on developing computational methods and more easily analyze the simulation results.

## 7. ACKNOWLEDGMENTS

Special thanks is given to Dr. P rsio L. A. Barros for his helpful suggestions and ideas for improving the system. The first author would also like to thank CAPES/Brazil for financial support.

## 8. REFERENCES

- [Bar99] Barros, P. L. A., and Mesquita, E. 1999. Elastodynamic Green's functions for orthotropic plane strain continua with inclined axes of symmetry. In: *International Journal of Solids and Structures*. 36, 31-32 (Nov. 1999), 4764-4788. DOI: 10.1016/S0020-7683(98)00264-9.
- [Gra91] Graff, K. F. 1991. *Wave motion in elastic solids*. Dover Publications Inc., Mineola, NY, USA. ISBN: 978-0-486-66745-4.
- [Lai09] Lai, W. M., Rubin, D., and Krempel, E. 2009. *Introduction to continuum mechanics*, 4th ed. Butterworth-Heinemann Ltd., Burlington, MA, USA. ISBN: 978-0-7506-8560-3.
- [Mes09] Mesquita, E., Adolph, M., Carvalho, E. R., and Romanini, E. 2009. Dynamic displacement and stress solutions for viscoelastic half-spaces subjected to harmonic concentrated loads using the Radon and Fourier transforms. In: *International Journal for Numerical and Analytical Methods in Geomechanics*. 33, 18 (Dec. 2009), 1933-1952. DOI: 10.1002/nag.802.
- [Raj93] Rajapakse, R. K. N. D., and Wang, Y. 1993. Green's functions for transversely isotropic elastic half space. In: *Journal of Engineering Mechanics*. 119, 9 (Sep. 1993), 1724-1746. DOI: 10.1061/(ASCE)0733-9399(1993)119:9(1724).

# Fast Temporal Filtering of Depth Maps

Sergey Matyunin  
Moscow State University  
Graphics & Media Lab  
smatyunin@graphics.cs.msu.ru

Dmitriy Vatolin  
Moscow State University  
Graphics & Media Lab  
dmitriy@graphics.cs.msu.ru

Maxim Smirnov  
YUVsoft Corp.  
ms@yuvsoft.com

## ABSTRACT

In this paper, we propose a method of filtering depth maps that are automatically generated from video sequences using optical flow, 3D reconstruction and scene analysis methods. To attain better quality, information from both the source video and depth map is used. The proposed algorithm uses motion estimation to take into account temporal information, but the algorithm's structure permits use of optical flow to improve quality, but at the expense of greater computation time. The method can be applied as a preprocessing stage for enhancement of multi-view or stereo video. Joint temporal and spatial processing can yield further improvements in quality. A comparison of the results with test ground-truth sequences using the BI-PSNR metric is presented.

## Keywords

Depth map, temporal filtering, stereo, 3D, video.

## 1 INTRODUCTION

Depth maps are widely used for 3D video production. Creating a depth map is a laborious process, so methods of automatic generation are under development. One of the promising approaches is depth map reconstruction using object motion [KMS07]. In [SCN05], the authors propose a method of spatial structure analysis based on neural networks and machine learning.

A relatively simple research direction involves calculating depth using stereo reconstruction [OA05]. Despite its simplicity, this approach also encounters many unsolved problems. Estimation of depth on the basis of stereo data aids parallax tuning for different types of screens and showing rooms, and it allows to take the parallax of neighboring scenes into account during nonlinear editing.

The problem of definite depth reconstruction without additional information is generally unsolvable. For automatic depth reconstruction, approaches that are based on local criteria minimization can be applied. This approach, however, leads to errors in the depth map. Such depth maps cannot be used for 3D image creation owing to temporal instability and errors. A specific type of preprocessing is required to

increase the temporal and spatial stability of the results. This paper proposes such a method of depth map processing using color and motion information.

## 2 RELATED WORK

Depth processing is often used to decrease the noticeability of depth map errors during visualization. Modified forms of Gaussian blur are applied in occlusion areas. In [TZ04], the authors propose asymmetric blurring: the filter length is larger in the vertical direction than in the horizontal direction. They also propose changing the size of the symmetric smoothing filter depending on the local values in the depth maps. An edge-dependent depth filter was proposed in [CCL<sup>+</sup>05]. To increase the quality of the results, edge direction is taken into account. Artifacts are more noticeable in occlusion areas. An adaptive method that responds to occlusions was proposed in [LH09].

The above-mentioned approaches only use data from the current frame, and they only use a portion of the color information from the source video (for example, only edges location).

A method of minimizing depth flickering for stationary objects was proposed in [KCKA10]; this method, however, only considers the presence of motion rather than the magnitude of the motion.

In [ZJWB09], the authors propose a method of reducing temporal instability by solving the energy minimization problem for several consecutive frames using graph cut and belief propagation. Another approach to depth map post-processing proposed in [ZJWB08] is iterative refinement. For each frame,

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.

the algorithm refines the depth maps of neighboring frames. The refinement procedure is also reduced to the energy minimization problem. Such approaches produce good results, but owing to computational complexity, they require a long time to process the entire video.

The proposed approach uses several neighboring frames to refine the depth map. Filtering is performed by taking into account the intensity (color) similarity of pixels and the spatial distance. The algorithm takes information about object motion into account using motion compensation.

### 3 PROPOSED METHOD

The proposed algorithm uses frames from the source video sequence along with a depth map generated from this video. We denote  $I_i(x)$  as the intensity (or color) of pixel  $x$  in frame  $i$ . This intensity is either a three-vector for a color image or a scalar for a grayscale image.  $n$  denotes the current frame number, and  $D_i(x)$  represents the depth for the  $i$ th frame in position  $x$ . The proposed method consists of four steps:

1. Motion estimation (ME) between the current frame  $I_n$  and neighboring frames  $I_{n+d}$ , where  $d = -m, \dots, -1, 1, \dots, m$ , and  $m > 0$  is a parameter. The result of this stage is a field of motion vectors  $MV_{n+d}(x)$ . We define  $MV_n(x) \equiv 0$ .
2. Computation of the confidence metric  $C_{n+d}(x)$  for the resultant motion vectors  $MV_{n+d}(x)$ . Here,  $C_{n+d}(x) \in [0, 1]$ .  $C_i(x)$  quantifies the estimation quality for motion vector  $MV_i(x)$ .
3. Motion compensation for the depth map and source frames. Here,  $D_{n+d}^{MC}$  denotes the motion-compensated depth maps, and  $I_{n+d}^{MC}$  denotes the motion-compensated source frames. Both the  $I_i^{MC}$  and  $D_i^{MC}$  images are computed using motion vectors  $MV_i$ , which are estimated from the source video sequence:

$$I_i^{MC}(x) = I_i(x + MV_i(x)),$$

$$D_i^{MC}(x) = D_i(x + MV_i(x)).$$

4. Depth map filtering using the computed  $D_i^{MC}$ ,  $C_i(x)$  and  $I_i^{MC}$  values.

#### 3.1 Motion Estimation

The results described in this paper were obtained using a block matching motion-estimation algorithm based on the algorithm described in [SGVP08]. We used macroblocks of size  $16 \times 16$ ,  $8 \times 8$  and  $4 \times 4$  with adaptive partitioning criteria. Motion estimation

is performed with quarter-pixel precision. Both luminance and chroma planes are considered. A confidence metric is calculated for quality assessment of the resultant motion vector field. The metric is similar to that described in [SGV08].

#### 3.2 Depth Filtering

We use  $2m + 1$  consecutive frames for filtering. The first step is temporal median filtering.

$$D_n^{med} = \underset{\substack{i=n-m, \dots, n+m \\ C_i > Th_C \\ |I_i^{MC} - I| < Th_{Diff}}}{med} D_i^{MC}.$$

The median is calculated over those pixels from the current and neighboring depth maps which have sufficiently small interframe difference  $|I_i^{MC} - I|$  and well-estimated motion vectors. Median filtering eliminates sharp discontinuities in the time domain.

The next processing step is temporal smoothing:

$$D_n^{smooth}(x) = \frac{\sum_{t=n-m}^{n+m} \sum_{y \in \sigma(x)} \omega(t, x, y) \cdot D_t^{input}(y)}{\sum_{t=n-m}^{n+m} \sum_{y \in \sigma(x)} \omega(t, x, y)},$$

where  $\omega(t, x, y)$  is a weight function, and  $D_t^{input}$  is the input depth map for this processing step. The source depth map  $D_i$  serves as the input depth map  $D_i^{input}$  for neighboring frames, and the filtered result  $D_n^{med}$  serves as the input  $D_n^{input}$  for the current frame. Previous fully processed depth maps  $D_{n-d}^{smooth}$  can be the input when processing current frame. This latter approach yields a smoother resulting depth map, but it is less accurate for small details.  $\sigma(x)$  denotes the spatial neighborhood of pixel  $x$ . The size of  $\sigma(x)$  involves a tradeoff between computation speed and processing quality. The weighting function  $\omega$  is

$$\omega(t, x, y) = f(|I_t^{MC}(y) - I_n(y)|) \cdot C_t(y) \cdot g(x, y),$$

where function  $f$  describes the dependence on the inter-frame difference;  $C_t(y)$  is the confidence for the motion compensation of pixel  $y$  in frame  $t$ ; and  $g$  denotes the dependence of weight on the spatial distance between  $x$  and  $y$ . In the simplest case,  $g$  is constant. To improve quality, we tested other relationships between the spatial distance and weight: linear, polynomial, and exponential. Function  $f$  is given by the formula

$$f(x) = \max \left( 0, \min \left( 1, \sum_{i=0}^3 \mu_i \cdot \left( \frac{x}{v} \right)^i \right) \right),$$

where  $\mu_i$  and  $v$  are parameters of the algorithm.

Thus, we average the depth value in the neighborhood of each pixel using information about interframe differences for the source video, the confidence metric for motion vectors, and spatial proximity.

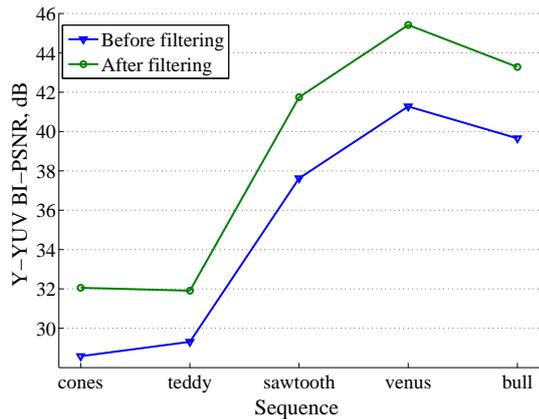


Figure 1: Results of the objective quality assessment. Depth maps were compared with ground truth depth before and after filtering. The comparison was performed using the Brightness Independent PSNR metric.

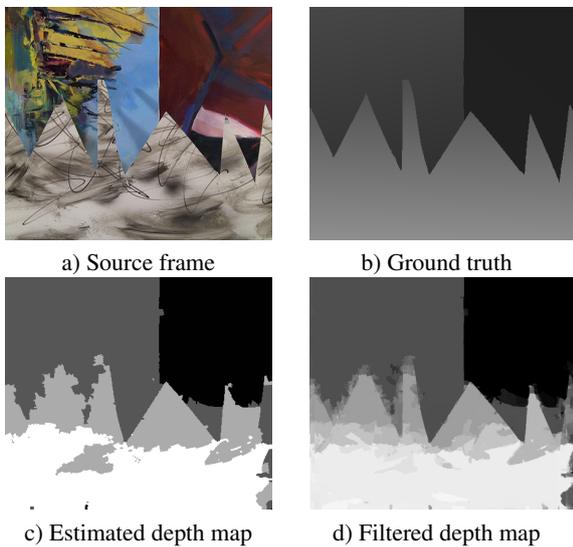


Figure 2: Comparison of results for "Sawtooth" sequence.

#### 4 RESULTS

The proposed algorithm was implemented in C as a console application. The algorithm uses one-pass processing, and it can be conveniently implemented in hardware. The algorithm's performance on an Intel Core2Duo T6670 processor running at 2.20 GHz is 7.6 fps for  $448 \times 372$  video resolution.

For an objective evaluation, the standard sequences "Cones," "Venus," "Sawtooth," "Teddy" and "Bull" were used [SS02, SS03]. The comparison with ground truth was performed using the Brightness Independent PSNR metric [VNG]. Fig. 1 shows the results. The source depth maps were obtained using the depth-from-motion method based on that described in [OA05].

For a subjective evaluation, Fig. 2 depicts the results of the algorithm. The depth map (Fig. 2b)

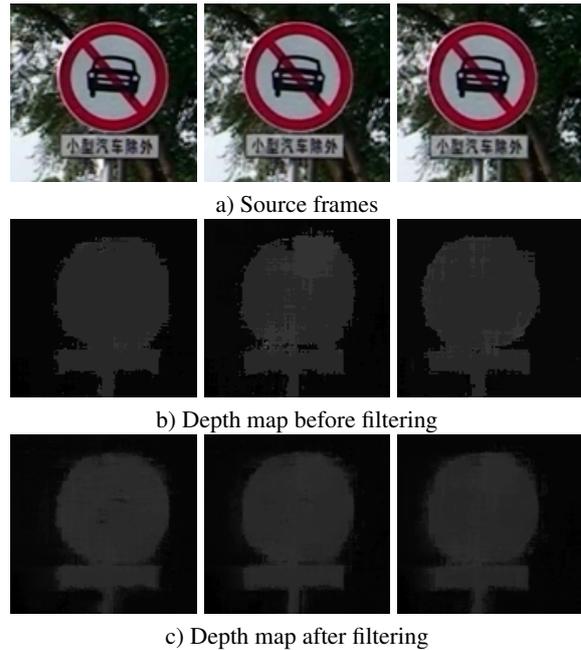


Figure 3: Segments of three consecutive frames for "Road" sequence.

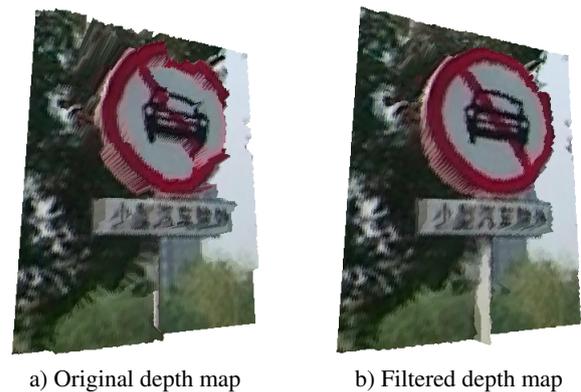


Figure 4: Depth-based rendered view for a segment of frame 112 for "Road" sequence. The view based on filtered depth seems more natural.

generated from the source video sequence (Fig. 2a) was filtered using the proposed method. The filtering process restored some lost details and fixed depth-estimation errors on object boundaries (Fig. 2d).

For the test sequence "Road" [ZJWB09], the proposed method improved the temporal stability of the depth map and recovered some details (see Fig. 3). The source depth map (Fig. 3b) has significant depth flickering. The proposed method improved the depth magnitude stability of objects in the scene (Fig. 3c).

We evaluated the visual quality of the rendered 3D image on the basis of an automatically generated depth map. The rendered view is shown in Fig. 4. The proposed method reduced artifacts in the resulting view and improved visual quality.

## 5 FURTHER WORK

In the proposed approach, we do not use occlusion detection and processing. In occlusion areas, motion estimation produces incorrect motion vectors, thus leading to artifacts. We intend to improve our confidence metric by implementing an algorithm for processing occlusion areas.

Better depth map quality can be achieved through spatial post-processing using information from the source video. Such filtering can be based on the assumption that uniform areas in the source video have uniform depth.

## 6 CONCLUSIONS

In this paper, we described a method of depth map filtering and presented a quality evaluation. The proposed algorithm improves the visual quality of depth maps and can simplify the manual work of 2D-to-3D video conversion. The described method allows the use of simpler and faster methods of automatic depth map generation without significant quality loss.

## ACKNOWLEDGEMENTS

This research was partially supported by grant number 10-01-00697-a from the Russian Foundation for Basic Research.

## REFERENCES

- [CCL<sup>+</sup>05] Wan-Yu Chen, Yu-Lin Chang, Shyh-Feng Lin, Li-Fu Ding, and Liang-Gee Chen. Efficient depth image based rendering with edge dependent depth filter and interpolation. *IEEE International Conference on Multimedia and Expo*, 0:1314–1317, 2005.
- [KCKA10] Sung-Yeol Kim, Ji-Ho Cho, Andreas Koschan, and Mongi A. Abidi. Spatial and temporal enhancement of depth images captured by a time-of-flight depth sensor. In *International Conference on Pattern Recognition (ICPR)*, pages 2358–2361. IEEE, 2010.
- [KMS07] D.H. Kim, D.B. Min, and K.H. Sohn. Stereoscopic video generation method using motion analysis. In *3DTV Conference*, pages 1–4, 2007.
- [LH09] Sang-Beom Lee and Yo-Sung Ho. Discontinuity-adaptive depth map filtering for 3d view generation. In *Proceedings of the 2nd International Conference on Immersive Telecommunications*, pages 1–6, 2009.
- [OA05] Abhijit S. Ogale and Yiannis Aloimonos. Shape and the stereo correspondence problem. *International Journal of Computer Vision*, 65(3):147–162, 2005.
- [SCN05] Ashutosh Saxena, Sung H. Chung, and Andrew Y. Ng. Learning depth from single monocular images. In *Advances in Neural Information Processing Systems 18*, pages 1161–1168. MIT Press, 2005.
- [SGV08] K. Simonyan, S. Grishin, and D. V. Vatin. Confidence measure for block-based motion vector field. In *GraphiCon*, pages 110–113, 2008.
- [SGVP08] Karen Simonyan, Sergey Grishin, Dmitriy Vatin, and Dmitriy Popov. Fast video super-resolution via classification. In *International Conference on Image Processing*, pages 349–352. IEEE, 2008.
- [SS02] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002.
- [SS03] Daniel Scharstein and Richard Szeliski. High-accuracy stereo depth maps using structured light. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:195, 2003.
- [TZ04] W. J. Tam and L. Zhang. Non-uniform smoothing of depth maps before image-based rendering. In *Proceedings of Three-Dimensional TV, Video and Display III (ITCOM'04)*, volume 5599, pages 173–183, 2004.
- [VNG] Dmitriy Vatin, Alexey Noskov, and Sergey Grishin. MSU Brightness Independent PSNR (BI-PSNR). [http://compression.ru/video/quality\\_measure/metric\\_plugins/bi-psnr\\_en.htm](http://compression.ru/video/quality_measure/metric_plugins/bi-psnr_en.htm).
- [ZJWB08] Guofeng Zhang, Jiaya Jia, Tien-Tsin Wong, and Hujun Bao. Recovering consistent video depth maps via bundle optimization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1 – 8, 2008.
- [ZJWB09] Guofeng Zhang, Jiaya Jia, Tien-Tsin Wong, and Hujun Bao. Consistent depth maps recovery from a video sequence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):974–988, 2009.

# POSTER: An Efficient Approach to Direct NURBS Surface Rendering for Ray Tracing

Aleksandrs Sisojevs

Riga Technical University, Department of  
images processing and computer graphics  
Meza Str. 1/3 – 304  
LV-1048, Riga, Latvia  
alexiv@inbox.lv

Aleksandrs Glazs

Riga Technical University, Department of  
images processing and computer graphics  
Meza Str. 1/3 – 338  
LV-1048, Riga, Latvia  
glaz@egle.cs.rtu.lv

## ABSTRACT

Ray tracing is a popular method for generating high quality images. NURBS surface is a popular tool for 3D modeling. There are different approaches to NURBS surface visualization by ray tracing methods. But these methods of computer graphics do not solve this task at an expected level. Therefore in this paper the authors propose to use a new effective approach for NURBS surface visualization by ray tracing method to render objects with high accuracy and quality.

## Keywords

NURBS, rendering, direct ray tracing.

## 1. INTRODUCTION

Most of the modern ray tracing based applications only deals with triangles as basic primitives. NURBS surface representation is common for most of 3D modeling tools because of its compactness and the useful geometric properties of NURBS surfaces. Using the direct raytracing of NURBS surfaces, one can achieve better quality of rendered images [Efr05a].

Martin et al. [Mar00a] describes a framework for ray-tracing of trimmed NURBS using Newton's method. A nonlinear equation system is used for finding the intersection point. The Newton's method is used for solving this equation system.

Nishita et al. [Nis90a] describes an iterative algorithm called Bézier Clipping, used to compute intersections between a ray and a Bézier patch by identifying and cutting away regions of the patch that are known not to intersect with the ray. There are several disadvantages in this approach, like numerical unstable work for some surfaces and

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.

problems in solving the case of multiple intersection points.

Efremov et al. [Efr05a] proposed some modification to Nishita's algorithm for Bézier and NURBS surfaces. But in this case, the number of patches and calculations increases.

Schollmeyer and Froehlich [Sch09a] describe an approach for NURBS surface ray tracing, where surface trimming is used to set of monotonic Bézier curves. For finding of intersection point the bisection method is used. But in this case, the number of calculations increases too.

This paper presents an effective approach for finding ray – NURBS surface intersection points, which are used for high-quality visualization of NURBS surfaces.

## 2. PROPOSED APPROACH

The mathematical task of finding an intersection point between the ray and a parametric surface can be described as a nonlinear equations system [Mar00a, Sis08a]:

$$\begin{cases} S'_x(u, v) = C_x(t) \\ S'_y(u, v) = C_y(t) \\ S'_z(u, v) = C_z(t) \end{cases}, \quad (1)$$

where:  $S'_x, S'_y, S'_z$  – surface equations,

$C_x(t), C_y(t), C_z(t)$  – ray equations.

A NURBS surface patch in Cartesian 3D space can be formulated as [Rog90b, Hea04b]:

$$S'(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m P'_{i,j} \cdot w_{i,j} \cdot N_{i,p}(u) \cdot N_{j,q}(v)}{\sum_{i=0}^n \sum_{j=0}^m w_{i,j} \cdot N_{i,p}(u) \cdot N_{j,q}(v)}, \quad (2)$$

where:  $P'_{i,j}$  – control points;  $w_{i,j}$  – weights;  $N_{i,p}(u)$ ,  $N_{j,q}(v)$  – the B-spline polynomials;  $p, q$  – the B-spline polynomials degree;  $u, v$  – parameters.

### Projection to $R^2$

Typically calculation is performed in  $R^4$  for rational patches [Nis90a]. Woodward [Woo89a] shows how the problem can be projected to  $R^2$ . This means that the number of calculations is reduced by 25%. This approach is used in [Nis90a] for rational Bezier patch subdivision. But this approach is good for NURBS surface calculation too. In this case the task of ray-surface intersection point search is transformed to the problem of non-linear equations system solving:

$$\begin{cases} S_X(u, v) - x_R = 0 \\ S_Y(u, v) - y_R = 0 \end{cases}, \quad (3)$$

where:  $S_X(u, v)$ ,  $S_Y(u, v)$  – are the surface equations on the projection plane,  $x_R, y_R$  – is the ray projection.

The system (3) solving task is divided into two parts: preprocessing and iterative root finding.

### Preprocessing

Preliminary searches for NURBS patch surface is equivalent to procedure for B-spline patch surfaces what is described in work [Sis09a]. In order to obtain the preliminary values of parameters  $u$  and  $v$  in every pixel, we propose to create a map of preliminary values. The map consists of the patch data and is coded in RGB channels. The Red channel contains the number of the patch. The Green and Blue channels contain the uniform gradient texture fill of the patches. The map is coded using the OpenGL graphics library. In this case the preliminary values of parameters can be described as follows:

$$\begin{cases} u_0 = (u_{\max} - u_{\min}) \cdot \frac{g}{255} + u_{\min} \\ v_0 = (v_{\max} - v_{\min}) \cdot \frac{b}{255} + v_{\min} \end{cases}, \quad (4)$$

where:  $u_0, v_0$  – parameters preliminary value in pixel;  $u_{\min}, u_{\max}, v_{\min}, v_{\max}$  – parameters  $u$  and  $v$  minimum and maximum;  $g, b$  – colour value in green and blue channels.

### Intersection Test

The Newton iteration [Tah03a] can be used for solving system (3) solving. In this case, iteration steps take the form:

$$\begin{bmatrix} u_{iter+1} \\ v_{iter+1} \end{bmatrix} = \begin{bmatrix} u_{iter} \\ v_{iter} \end{bmatrix} + \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix}, \quad (5)$$

where:

$$\begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} = -[J]^{-1} \cdot [F], \quad (6)$$

where:

$$[J] = \begin{bmatrix} \frac{\partial S_X}{\partial u} & \frac{\partial S_X}{\partial v} \\ \frac{\partial S_Y}{\partial u} & \frac{\partial S_Y}{\partial v} \end{bmatrix}; [F] = \begin{bmatrix} S_X - x_R \\ S_Y - y_R \end{bmatrix}, \quad (7)$$

$[J]$  is Jacobian matrix.

### Proposed method

The NURBS patch surface equation on the projection plane can be described as follows:

$$S(u, v) = \left( \frac{x}{w} \quad \frac{y}{w} \right), \quad (8)$$

where:  $x, y$ , and  $w$  – are numerator and denominator of NURBS surface equations on the projection plane.

Using NURBS surface partial derivatives [Rog90b], after transformation, the equation (6) takes the form:

$$\begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} = \frac{w}{M} \cdot \begin{bmatrix} M_U \\ M_V \end{bmatrix}, \quad (9)$$

where:

$$M = \begin{vmatrix} x & \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ y & \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \\ w & \frac{\partial w}{\partial u} & \frac{\partial w}{\partial v} \end{vmatrix}; M_U = \begin{vmatrix} x & x_R & \frac{\partial x}{\partial v} \\ y & y_R & \frac{\partial y}{\partial v} \\ w & 1 & \frac{\partial w}{\partial v} \end{vmatrix}, \quad (10)$$

and

$$M_V = \begin{vmatrix} x & \frac{\partial x}{\partial u} & x_R \\ y & \frac{\partial y}{\partial u} & y_R \\ w & \frac{\partial w}{\partial u} & 1 \end{vmatrix}, \quad (11)$$

In contradistinction to method Martin et al. [Mar00a], where both the rational and non-rational equations partials derivatives were calculated, in the proposed method only the non-rational equations partials derivatives was calculate. This approach allows to decrease number of calculations and speed-up rendering time.

### Termination Criteria

In this work, three criteria are used to decide when to terminate the Newton iteration. There are drawn [Yang87a]. The first condition is success criterion: if we are closer to the root than some predetermined  $\varepsilon_1$ :

$$\|F(u_{iter}, v_{iter})\|_2 < \varepsilon_1, \quad (12)$$

In this case, we report an intersection to. Otherwise, we continue the iteration. There is no allowing the new  $(u_{iter}, v_{iter})$  estimate to take us farther from the root than the previous one:

$$\|F(u_{iter}, v_{iter})\|_2 > \|F(u_{iter-1}, v_{iter-1})\|_2, \quad (13)$$

In this work check is made to assure that the matrix  $[M]$  is not singular:

$$|\det(M)| < \varepsilon_2, \quad (14)$$

### 3. EXPERIMENTAL RESULTS

In this work the proposed method, as well as the methods suggested by Martin et al. were implemented. In order to visualize a scene the 1 ray/pixel approach was used. The size of the images in experiment is 512\*512 pixels. The experiments were carried out on a computer with CPU Intel Xeon 3,2 GHz, RAM 2 GB. The received images are shown in Figures 1-6. Image rendering time is shown in Table 1. For comparison we shall consider the time of visualization in percentage, by taking earlier known method (Martin et al.) for 100%.

Objects	Proposed method, sec.	Martin et al. Method, sec	Speed-up of rendering, %
“Mobile phone”	4,297	5,422	20,75
“Machine component”	1,890	2,203	14,21
“27 Ducks”	9,0	11,86	24,11

Table 1. Image rendering time

### 4. CONCLUSION

In this work an efficient approach to direct NURBS surface rendering for ray tracing is proposed. The results (in Figures 1 – 6) shows, that:

- As it is possible to see from these figures the proposed method gives an advantage on quality of the images (there’s no distortion on the borders of patches).
- As seen from the table, the proposed method gives stable results in the fastest rendering

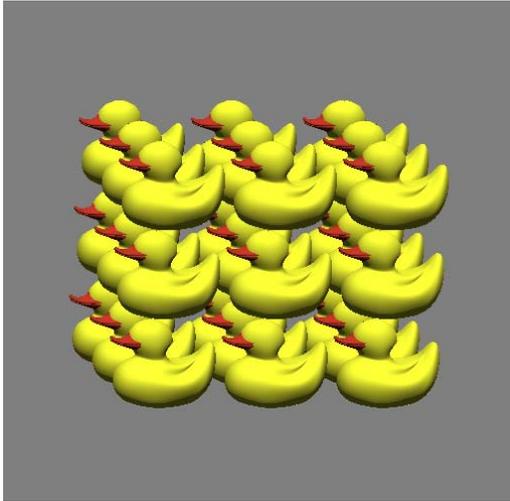
time in our experiments (speed-up is 14% – 24% than the method of Martin et al.).

### 5. ACKNOWLEDGMENTS

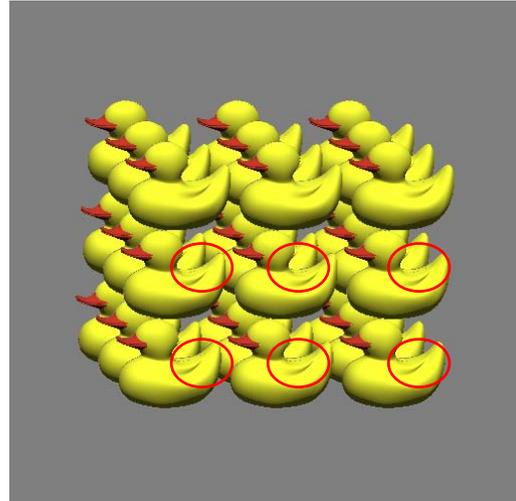
This work has been supported by the European Social Fund within the project «Support for the implementation of doctoral studies at Riga Technical University».

### 6. REFERENCES

- [Efr05a] Efremov, A., Havran, V. & Seidel, H.-P. ‘Robust and numerically stable Bézier clipping method for ray tracing NURBS surfaces.’ In Proceedings of the 21st Spring Conference on Computer Graphics SCCG’05:127 - 135. 2005
- [Hea04b] Hearn, D. and Baker, M.P. Computer Graphics with OpenGL, Prentice Hall, 2004.
- [Him72b] Himmelblau, D. Applied Nonlinear Programming, – McGraw-Hill Book, 1972.
- [Mar00a] Martin, W., Cohen, E., Fish, R., & Shirley, P. (2000). ‘Practical Ray Tracing of Trimmed NURBS Surfaces.’ JGT 5(1): 27-52. 2000.
- [Nis90a] Nishita. T., Sederberg. T.W., & Kakimoto M. Ray tracing trimmed rational surface patches, Computer Graphics, vol.24, no. 4, pp. 337–345, 1990.
- [Rog90b] Rogers, D.F. and Adams, J.A. Mathematical Elements for Computer Graphic, 2nd Ed., McGraw-Hill, Boston, MA, 1990.
- [Sch09a] Schollmeyer A., Froehlich, B. Direct Trimming of NURBS Surfaces on the GPU, ACM Transactions on Graphics, Volume 28, Issue 3, Article No. 47, 2009.
- [Sis08a] Sisojevs A., Glazs A. An new approach of visualization of free-form surfaces by a ray tracing method, IEEE MELECON Proceedings, 2008, pp 872-875
- [Sis09a] Sisojevs A., Glazs A. POSTER: Efficient approach to direct B-spline surface rendering by a ray tracing, The International Conference WSCG’2009 Poster proceedings, 2009. - 13- 16 p.
- [Tah03b] Taha H.A. Operations Research: an Introduction, 7th Ed. – New Jersey: Prentice Hall, 2003. – 830 p.
- [Woo89a] Woodward C. Ray tracing parametric surfaces by subdivision in viewing plane// Theory and practice of geometric modeling book contents, Springer-Verlag New York, Inc., New York, NY, USA. – 1989. pp. 273 – 287.
- [Yan87a] Yang C.-G. On speeding up ray tracing of B-spline surfaces. Computer Aided Design, 19(3), April 1987.



**Figure 1.** The image of a scene “27 Ducks” obtained using the proposed method.



**Figure 2.** The image of a scene “27 Ducks” obtained using the method Martin et al.



**Figure 3.** The image of an object “Mobile phone” obtained using the proposed method.



**Figure 4.** The image of an object “Mobile phone” obtained using the method Martin et al.



**Figure 5.** The image of an object “Machine component” obtained using the proposed method.



**Figure 6.** The image of an object “Machine component” obtained using the method Martin et al.

# Comparison of Mean Shift Algorithms and Evaluation of Their Stability

Stepan Srubar

Department of Computer Science, FEECS,  
VŠB - Technical University of Ostrava,  
17. listopadu 15, 708 33 Ostrava, Czech Republic  
stepan.srubar@vsb.cz

Milan Surkala

Department of Computer Science, FEECS,  
VŠB - Technical University of Ostrava,  
17. listopadu 15, 708 33 Ostrava, Czech Republic  
milan.surkala@vsb.cz

## ABSTRACT

Mean Shift is an universal and robust segmentation algorithm used also for image segmentation. Implementation corresponding to definition is slow for real-time image processing, thus some faster variants were proposed. Speed and quality of the segmentation can be influenced by the variant of the algorithm as well as by setting its parameters. Modification of a parameter causes change of number of segments and their shape as well. By using segmentation evaluation algorithm, we analyze and present these changes for original and modified Mean Shift algorithms. Influence of noise is also studied.

## Keywords

Mean Shift, segmentation stability, Segmentation Difference.

## 1. INTRODUCTION

Mean Shift is a segmentation method for point data originally. By representing color channels of an image as other dimensions, we could use Mean Shift for segmentation of images [Com02a]. The main disadvantage lies in slow computation. Therefore, some variants of original Mean Shift algorithm were proposed.

We analyze original and modified algorithms by using evaluation method called Segmentation Difference [Sru10a]. Unlike other methods for evaluation of segmentations, this can evaluate refinement of a segmentation separately from the shape of common objects. By changing parameters of a segmentation algorithm, resulting segmentations should differ in refinement, not in the shape of borders of segments.

Similar evaluation was recently presented [Kaf08a]. By using a simple evaluation method, they were restricted to comparison of segmentations created by different algorithms but with the same parameters.

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.

We are able to analyze influence of different algorithms as well as different parameters.

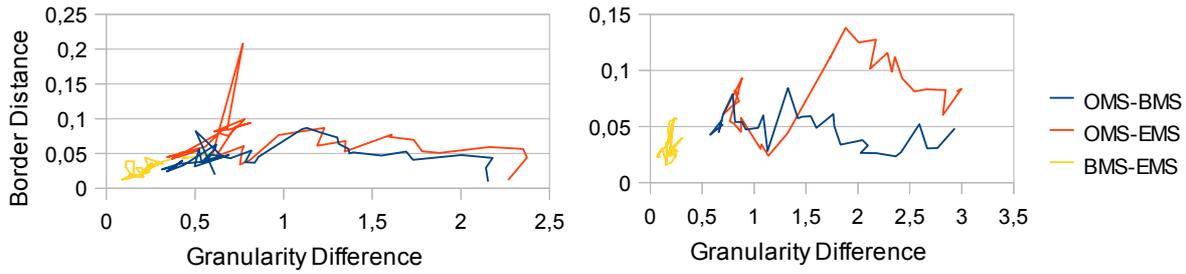
## 2. MEAN SHIFT ALGORITHMS

### Original Mean Shift

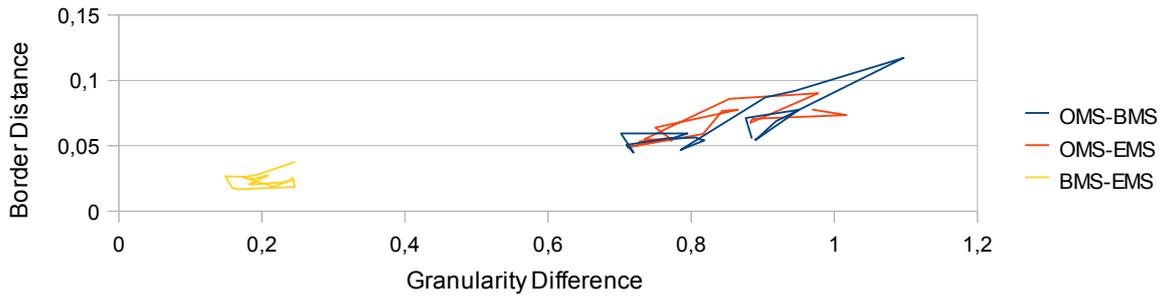
Segmentation by Mean Shift is based on density of points only. For each point, path of the steepest density increase for the point is found. End of each path is located in some local density maximum (called attractor). All pixels having the same attractor are put to common segment. The path is computed iteratively:

$$x^{(i+1)} = \frac{\sum_j k\left(\left\|\frac{x^{(i)} - x_j}{h}\right\|\right) \cdot x_j}{\sum_j k\left(\left\|\frac{x^{(i)} - x_j}{h}\right\|\right)},$$

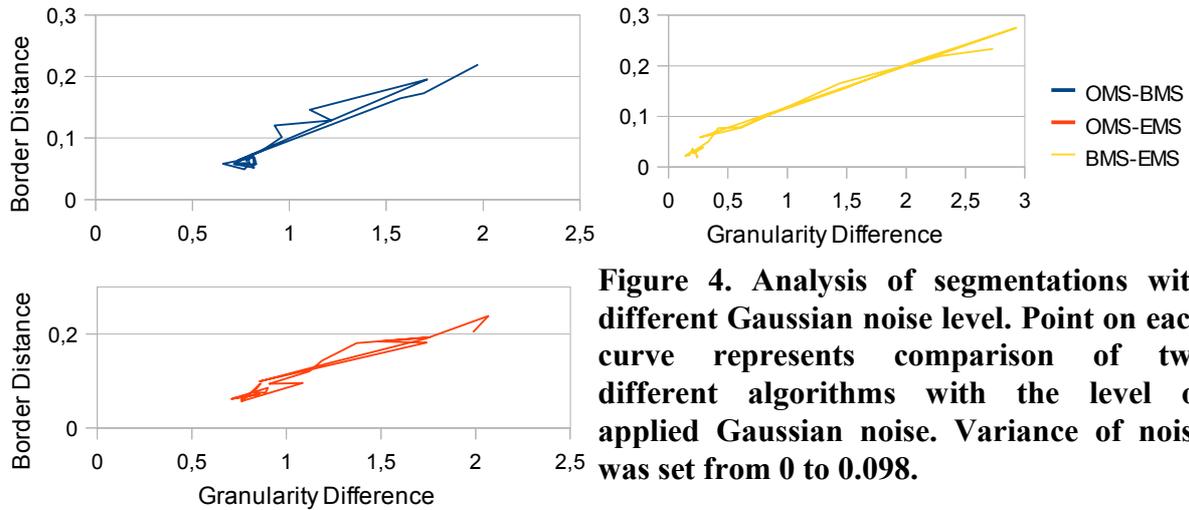
where  $x^{(i)}$  is location on the path in iteration  $i$ ,  $x_j$  is a location of a data point and  $k(y)$  is a weighting function denoted as a kernel,  $h$  is a vector and the division is point-wise. Typically, the kernel is non-zero only on interval  $\langle 0,1 \rangle$  and vector  $h$  determines the size of the influence of the kernel in each dimension separately. In further comparisons, we will denote this original Mean Shift as OMS.



**Figure 2. Analysis of segmentations with different spatial (left) and color (right) kernel sizes. Point on each curve represents comparison of two different algorithms with the same parameters. Size of spatial kernel size was set from 0 to 40. Size of color kernel size was set from 0 to 1.**



**Figure 3. Analysis of segmentations with different kernel shapes. Exponents were set from 0.2 to 10.**



**Figure 4. Analysis of segmentations with different Gaussian noise level. Point on each curve represents comparison of two different algorithms with the level of applied Gaussian noise. Variance of noise was set from 0 to 0.098.**

### Blurring Mean Shift

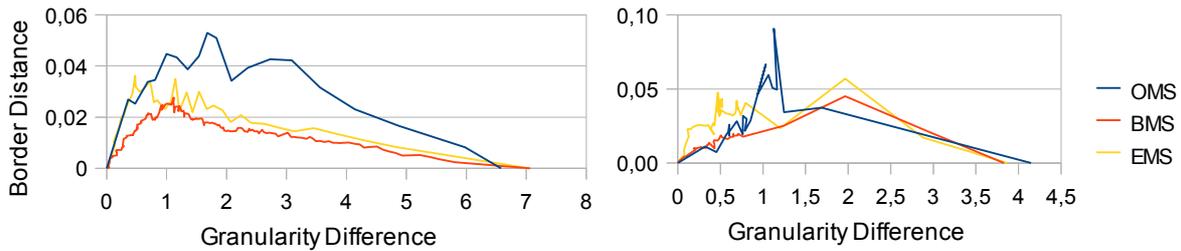
Mean Shift variant called Blurring Mean Shift (BMS) was presented in [Car00a, Car06a]. BMS formula is given by

$$x^{(i+1)} = \frac{\sum_j k\left(\left\|\frac{x^{(i)} - x_j^{(i)}}{h}\right\|\right) \cdot x_j^{(i)}}{\sum_j k\left(\left\|\frac{x^{(i)} - x_j^{(i)}}{h}\right\|\right)},$$

where the term  $x_j^{(i)}$  represents point from the previous iteration.

### Evolving Mean Shift

In 2009 whole new idea of Mean Shift segmentation was developed [Zha09a] called Evolving Mean Shift (EMS). It is an energy minimization method. EMS computes mean shift vectors for all pixels in an image. Then EMS searches for the longest mean shift vector and proceeds the shift of the corresponding pixel. Therefore, the energy is maximally decreased. This is repeated until the energy decreases to some level. Energy of a data set can be written as



**Figure 5. Analysis of stability of methods according to spatial kernel size. Smaller kernels than in reference segmentation are on the left graph, larger kernels are on the right graph. Size of spatial kernel size was set from 20 to 0 (using steps of size 0.1) and from 20 to 40 (steps 0.5).**

$$E_{x_i} = \sum_{j \neq i} (E_{x_i x_j} + E_{x_j x_i}),$$

$$E_{x_i x_j} = k \left( \left\| \frac{x_i - x_j}{h} \right\| \right).$$

### 3. COMPARISON

All segmentations are created using gray-scale image of Lena with resolution 256 x 256 pixels. Each point on a curve in figures 2-4 represents comparison of two segmentations with the same parameters by Segmentation Difference. Each graph covers influence of change of a single parameter, while the other parameters are preserved.

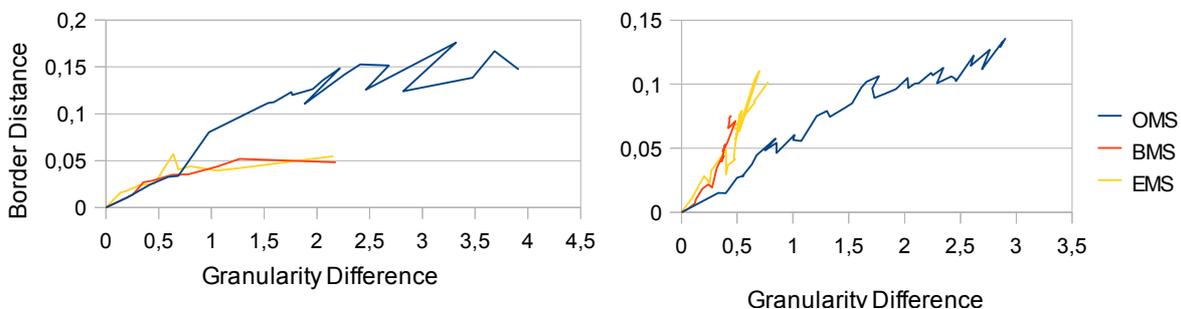
Reference elements of vector  $h$  corresponding to spatial dimensions are set to 20, elements of color space are 0.125. An image is internally represented in the range  $\langle 0,1 \rangle$ . Stopping conditions are 200 iterations or minimal change of position 0.001. Reference kernel is Epanechnikov kernel:

$$k(y) = 1 - y^2.$$

#### Kernel size

Kernel function is non-zero only on interval  $\langle 0,1 \rangle$  but we change an area of influence of the kernel by setting of elements of the vector  $h$ , which will be denoted naturally but inexactly as a kernel size.

Figure 2 represents comparisons for different spatial kernel sizes as well as for different color kernel sizes.



**Figure 6. Analysis of stability of methods according to color kernel size. Smaller kernels than in reference segmentation are on the left, larger kernels are on the right. Size of color kernel was set from 0 to 1.**

Similar segmentations are placed near to 0. Evidently, BMS and EMS are very similar in this test.

#### Kernel shape

Kernel is, typically, a decreasing function. We are not able to compare all decreasing functions, thus we analyze kernel types defined as follows

$$k(y) = 1 - y^z, z \in \langle 0, \infty \rangle.$$

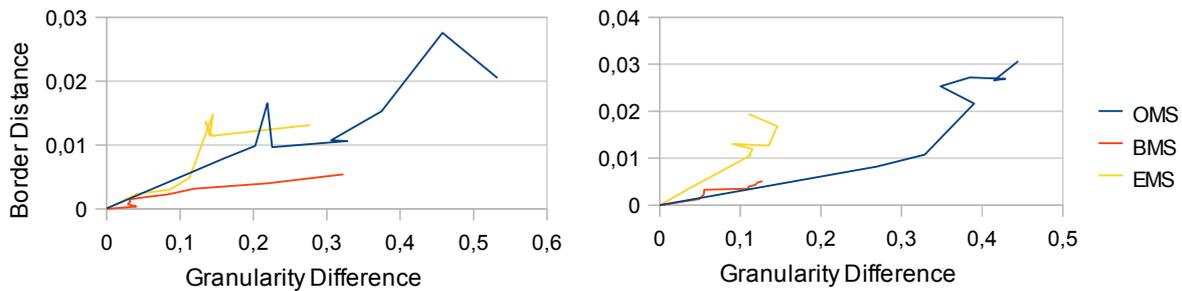
One extreme case where  $k(y)=1$  is called uniform kernel. By setting  $z$  to 1 and 2 gives us another widely used triangular and Epanechnikov kernel respectively. We used exponents from 0.2 to 10. Analysis of influence of the kernel shape on the resulting segmentation can be seen in the figure 3.

#### Influence of Noise

Our approach consists of adding Gaussian noise with variance from 0 to 0.098 to the reference image (fig. 1) and measuring differences between all three variants of Mean Shift (see figure 4).

### 4. ANALYSIS OF STABILITY

Segmentation algorithm is, typically, influenced by its parameters. By changing parameters, we could expect change in number of segments, not change of shape or position of segments. Such preservation of shape and position will be called stability of a segmentation algorithm.



**Figure 7. Analysis of stability of methods according to shape of the kernel. Results with kernels with exponent from 2 to 0.2 are on the left, the rest from 2 to 10 is on the right.**

Stable segmentation would have all points lying on the horizontal axis. First stability analysis graphs in figure 5 represent influence of spatial size of the kernel. Next couple of graphs in figure 6 analyze influence of the color kernel size. Influence of the kernel shape is presented in the figure 7. Last graphs in the figure 8 tries to analyze influence of the Gaussian noise on the resulting segmentation.

## 5. DISCUSSION

Figures 2 and 3 shows some similarity of BMS and EMS. In other words, their segmentations look nearly the same for various kernel shapes and sizes. But any influence of noise leads to different segmentations of these methods. Original Mean Shift segmentation algorithm is much more unstable when changing its parameters in comparison with BMS and EMS. That is evident from high results of Border Distance values. However, OMS is the most stable algorithm in the presence of a noise. According to these evaluations, we cannot easily select the best algorithm.

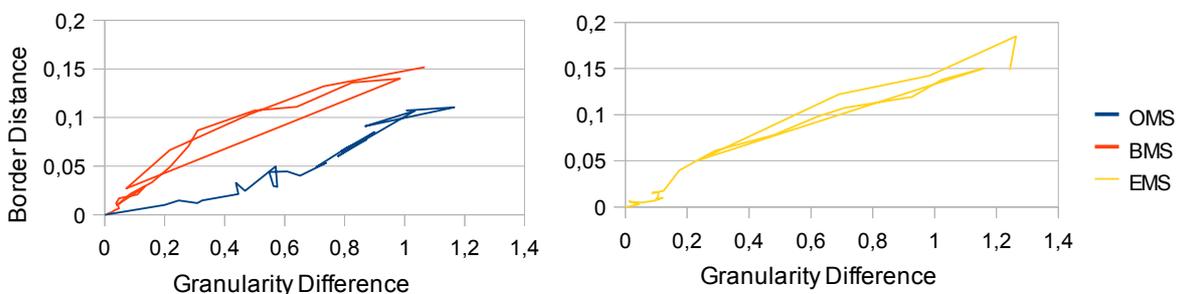
## 6. CONCLUSION

We implemented three variants of Mean Shift algorithm. We compared them to each other and tested their stability according to change of some of their parameters and change of variance of noise. None of them can be marked as the best one, still we

found high correlation between BMS and EMS variants.

## 7. REFERENCES

- [Car00a] Carreira-Perpinan, M. A. Mode-finding for mixtures of Gaussian distributions. Technical report. University of Sheeld, 2000.
- [Car06a] Carreira-Perpinan, M. A. Fast Nonparametric Clustering with Gaussian Blurring Mean-Shift. Proceedings of the 23rd international conference on Machine learning, vol. 148, 2006.
- [Com02a] Comaniciu, D., and Meer, P. Mean shift: A robust approach towards feature space analysis. IEEE Transactions on pattern analysis and machine intelligence, vol. 24, No. 5, May 2002.
- [Kaf08a] Kaftan, J.N., Bell, A.A., and Aach, T. Mean Shift Segmentation Evaluation of Optimization Techniques. In proceedings of the Third International Conference on Computer Vision Theory and Applications, pp. 365-374, VISAPP 2008.
- [Sru10a] Srubar, S. Toward Objective Segmentation Evaluation. International Workshops on Computer Graphics, Vision and Mathematics, 2010.
- [Zha09a] Zhao, Q., Yang, Z., Tao, H. Evolving Mean Shift with Adaptive Bandwidth: A Fast and Noise Robust Approach. Computer Vision – ACCV 2009, Springer Berlin / Heidelberg, 2010.



**Figure 8. Analysis of stability of methods according to Gaussian noise level. Variance of noise was set from 0 to 0.098.**

# Generation of Indian Ink Painting Image from Two-Dimensional Image Data

Kenichi Kobori

Osaka Institute of Technology  
1-79-1 Kitayama  
Hirakata, Osaka, 573-0196 Japan  
kobori@is.oit.ac.jp

Naomi Maekawa

Osaka Institute of Technology  
1-79-1 Kitayama  
Hirakata, Osaka, 573-0196 Japan

Koji Nishio

Osaka Institute of Technology  
1-79-1 Kitayama  
Hirakata, Osaka, 573-0196 Japan  
nishio@is.oit.ac.jp

## ABSTRACT

Recently, non-photorealistic image rendering (NPR) has become the subject of research and development by many researchers due to a rapid improvement in computer graphics. NPR has been used for communication by emphasizing an image or to simulate existing painting techniques. In this paper, we propose a generation method for Indian ink painting image which is an Oriental painting technique from two-dimensional realistic image. In general, Indian ink painting has three kinds of drawing techniques, "Senbyo" technique to draw only contours of objects, "Mokkotu" technique to draw only the interior of objects and "Senzen" technique to draw both contours and the interior of objects. In this paper, we simulate Indian ink painting by "Senzen" technique. We extract the contours from two-dimensional realistic image and decide the interior of the object in order to generate the image which is drawn by "Senzen" technique. The final image is generated by combining the inner area of the object with the contours. We generated several Indian ink painting images by our method. The experimental results show that the proposed method is effective for Indian ink painting image rendering.

## Keywords

NPR, Indian ink painting, bilateral filter, Fourier transform

## 1. INTRODUCTION

The driving force behind computer graphics has been photorealism until now. Recently, NPR has been active area of research with most of the work concentrating on generating images in various traditional styles. NPR is classified in a method for communication and a method to express existing painting techniques. Many methods to express existing painting techniques are proposed. Haeberli [ Haeberli ] describes a stroke based algorithm using the haloed line effect. Kim et al. [ Kim ] describe a method by pointillism. Kang et al. [ Kang ] describe a method of generating many kinds of images, such as oil painting, watercolor painting and stripping, by changing the feature of the stroke. However, Indian ink painting has not received as much focus. Satoh et al. [ Satoh ] give a method of generating Indian ink painting.

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, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

It is necessary to use a three dimensional object as an input data to their method. Therefore, we propose a method for generating Indian ink painting from a two-dimensional realistic image. In general, Indian ink painting has three kinds of drawing techniques, "Senbyo" technique to draw only contours of objects, "Mokkotu" technique to draw only the interior of objects and "Senzen" technique to draw both contours and the interior of objects. In this paper, we simulate Indian ink painting by "Senzen" technique.

## 2. PROPOSED METHOD

We generate Indian ink painting by giving several filter processings against two-dimensional objects in an input image. After smoothing input image by bilateral filter, we extract the contours and decide the interior of the object in order to generate the image which is drawn by "Senzen" technique. The final image is generated by combining the interior with the contours. A flow of the method is shown in Figure 1.

### 2.1 Smoothing

The features of Indian ink painting are as follows;

- 1) Line is drawn along a boundary of the object by a painting brush.

- 2) The light and shading of India ink changes smoothly along the boundary.
- 3) There are few blurs of India ink near the boundary of the object and the boundary is clear.

Based on these features, we use bilateral filter [ Tomasi ],etc. along the boundary in order to smooth the image while maintaining contours of the object.

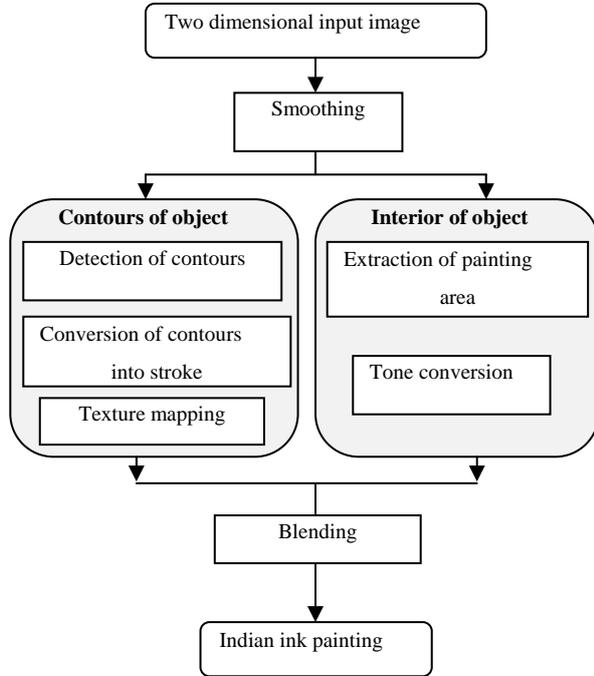


Figure 1. A flow of the method.

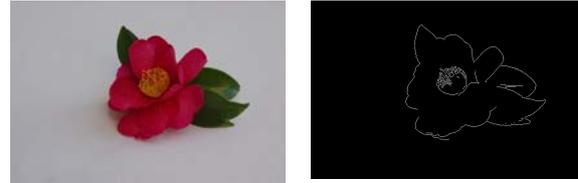
First, the input image is converted to a gray scale image. We calculate the gradient vector of every pixel of the gray scale image and the tangent vector that is perpendicular to the vector by using sobel filter. Using the tangent vector, bilateral filter is applied to the gray scale image along the boundary of the object. The weight of Gaussian distribution  $\sigma_g$  of the gradient vector direction in every pixel is calculated using the gradient vector  $\mathbf{g}$ , as follows;

$$\sigma_g(x, y) = (1 - g(x, y))\sigma_t \quad (1)$$

where  $\sigma_t$  is the weight of Gaussian distribution of tangent vector direction.

## 2.2 Detection of object contours

We apply Laplacian filter to the image which was smoothed and extract the pixels to constitute edges by image thresholding. Figure 2 shows an example of the input image and the boundary image.



(a) Input image; (b) Binary image.  
Figure 2. Edge extraction:

Extracted pixels are converted to poly line data by vector trace method [Agui] and B-spline stroke data are generated by vertices of the ploy line data. We map the brush texture data shown in Figure 3 on the stroke data by texture mapping technique. As a result, we get the image shown in Figure 4.



Figure 3. Texture data of contour.



Figure 4. The result of texture mapping.

## 2.3 Extraction of painting area

The interior of the object is painted with light Indian ink in “Senzen” technique. It is necessary to extract the interior of the object to be painted. Our method extracts the area in the smoothed image by frequency filtering. The image is converted into spatial frequency domain image by Fourier transform. In general, the light and dark shading near the contour is thicker than the shading of the inner part in Indian ink painting.

Therefore, we emphasize middle frequency of spatial frequency domain image and remove the low frequency using the equations;

$$H(u, v) = H_{high}(u, v) + H_{b-emph}(u, v) \quad (2)$$

$$H_{high}(u, v) = 1 - G(u, v, \sigma_1) \quad (3)$$

$$H_{b-emph}(u, v) = N(G(u, v, \sigma_2)G(u, v, \sigma_3)) \quad (4)$$

$$G(u, v, \sigma) = \exp\left(-\frac{u^2 + v^2}{2\sigma^2}\right) \quad (5)$$

where  $u$  is the frequency component of the X axis direction,  $v$  is the frequency component of the Y axis

direction, and  $N$  is a function to normalize a value to 1 from 0,  $\sigma_1$  is the variance of high-pass filter, and  $\sigma_2, \sigma_3$  are the variances to emphasize the light and dark shading near the contour.

Indian ink painting has the characteristic that the part of low brightness is painted with Indian ink. We reverse the sign of frequency before inverse Fourier transform process is executed in order to express this characteristic. We get the Indian ink painting image for the interior of the object shown in Figure 5 by reversing the image.



Figure 5. Indian ink painting for the interior.

## 2.4 Tone conversion

Indian ink painting has the characteristic that we sometimes paint the interior of the object over again by the different light and dark shade. The inner part of red circle in Figure 6 shows an example of this characteristic.



Figure 6. Expression of the area of an object.

We emphasize the boundary of light and dark shade by using tone conversion function [Winnemoller] in equation (6)-(8) in order to express this characteristic. Converted image  $R(x,y)$  is calculated as follows;

$$R(x, y) = q_{near} + \frac{\Delta q}{2} \tanh \left( \varphi \left\{ I(x, y) - \left( q_{near} + \frac{\Delta q}{2} \right) \right\} \right) \quad (6)$$

$$q_{near} = \frac{[(q-1)f(x, y)]}{q-1} \quad (7)$$

$$\Delta q = \frac{1}{q} \quad (8)$$

where  $I(x,y)$  is the image before conversion,  $\varphi$  is the parameter of smoothness of tone boundary, and  $q$  is the number of the tone for conversion.

For example, the image of the result in Figure 7 is generated when we use the function in Figure 8 ( $q = 5, \varphi = 30$ ).



Figure 7. Converted image.

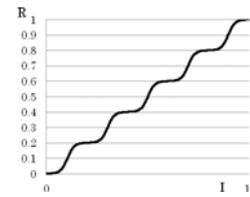


Figure 8. Conversion function.

## 2.5 Blending of image

We compose the image of the contour generated in 2.2 and the image of the interior generated in 2.3 by using alpha blending. The final image  $O(x,y)$  is calculated by equation (9) and (10).

$$O(x, y) = \alpha(x, y)T(x, y) + (1 - \alpha(x, y))R(x, y) \quad (9)$$

$$\alpha(x, y) = 1 - T(x, y) \quad (10)$$

where  $T$  is the image of the contour generated in 2.2 and  $R$  is the image of the interior generated in 2.3. Figure 9 shows the image that blends Figure 4 and Figure 7.

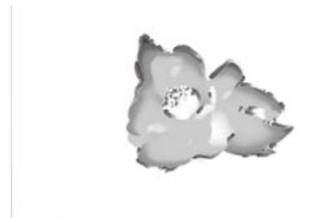


Figure 9. Blended image.

## 3. EXPERIMENTS AND RESULTS

We experiment to verify the effectiveness of our method.

Figure 10 shows three kinds of two-dimensional image that we use as input images. There is one object in each image, Japanese persimmon, leaf and Japanese maple.

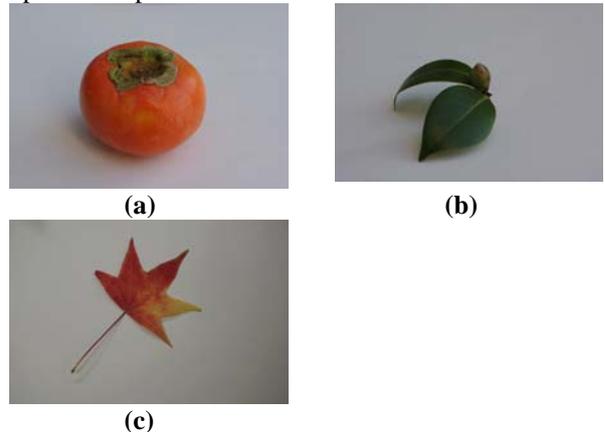
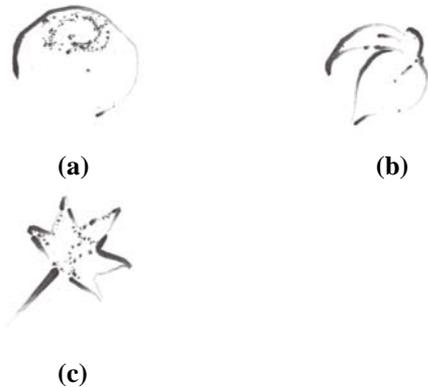


Figure 10. Input images:

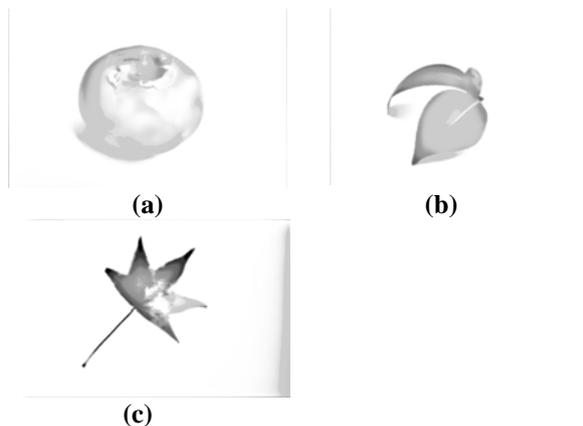
(a) Persimmon; (b) Leaf; (c) Japanese maple.

Figure 11 shows the results of contour painting and Figure 12 shows the results of interior painting. The

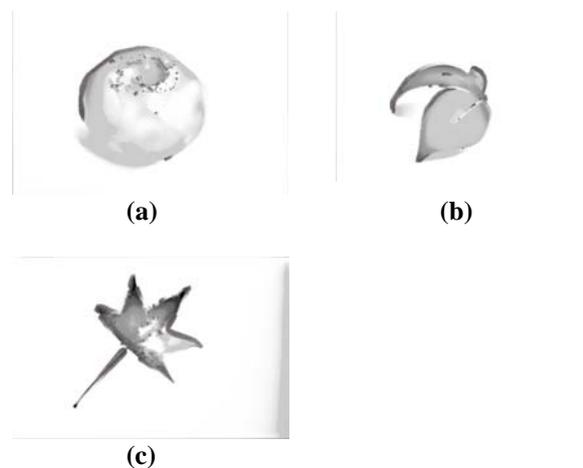
final blending image shows in Figure 13. The contours of Indian ink painting are expressed on the boundary of the object in Figure 11 and light and dark shading near the contour is thicker than the shading of the inner part shown in Figure 12.



**Figure 11. The result of contour painting:**  
(a)Persimmon; (b)Leaf; (c)Japanese maple.



**Figure 12. The result of area painting:**  
(a)Persimmon; (b)Leaf; (c)Japanese maple.



**Figure 13. Blended images:**  
(a)Persimmon; (b)Leaf; (c)Japanese maple.

## 4. CONCLUSION

In this paper, we proposed an automatic generation method of Indian ink painting image from a two-dimensional image. Our method generates two kinds of Indian ink painting images to express the painting by “Senzen” technique. One is contour painting of the object and the other is interior painting of the object. We achieved Indian ink painting image by blending these two kinds of images.

Experimental results showed that our method is effective.

Future work will explore the automatic generation of the tone function. The values of  $q$  and  $\phi$  in equation (6)-(8) depend on the input image. In this study, we decided these values through trial and error. It is necessary to conceive a method to calculate the values of  $q$  and  $\phi$  automatically according to the input image.

## 5. REFERENCES

- [Haerberli]Paul Haerberli: Paint By Numbers:Abstract Image Representations. Proc. of SIGGRAPH 1990, pp.207-214, 1990.
- [Kim]Sung Y. Kim,Ross Maciejewski, Tobasis Isenberg, William M. Andrews, Wei Chen, Maro C.Sousa: Stipping by Eample. Proc. of Npar 2009, pp.42-50, 2009.
- [Kang]Hyung W. Kang, Charles K. Chui, Uday K.Chakaraborty: Unified Scheme for Adaptive Stroke-Based Rendering. The Visual Computer, Vol.22, No.9, pp.814-824, 2006.
- [Satoh]Youetsu Satoh, Tadahiko Fujimoto, Kazunobu Muraoka, Norishige Chiba: Stroke-Based *Suibokuga*-Like Rendering for Three-Dimensional Geometric Models-*Ten* and *Shun* Touches. The Journal of the society for Art and Science, Vol.3, No.4, pp.224-234, 2004.
- [Tomasi]Carlo Tomasi,Roberto Manduchi: Bilateral Filtering for Gray and Color Images. International Conference on Computer Vision, pp.839-846, 1991.
- [Agui] Takeshi Agui, Hisato Iitsuka, Masayuki Nakajima: Image Processing for Urban Map Utilizing Pyramid Hierarchical Data. The Institute of Electronics, Information and Communication Engineers, Vol.J65-D No.10, pp.1243-1249,1982.
- [Winnemoller]Holger Winnemoller,Sven C. Olsen, Bruce Gooch: Real-Time Video Abstraction. Proc. of SIGGRAPH 2006, pp.1221-1226, 2006.

# Deformation Exchange between Adjacent Physical Code Geometries

B. Duplex<sup>(1,3)</sup> G. Gesquière<sup>(3)</sup> M. Daniel<sup>(3)</sup> F. Perdu<sup>(2)</sup>

<sup>(1)</sup> CEA, DEN  
DER/SSTH/LDAL  
Cadarache  
Saint-Paul-lez-Durance  
F-13108, France

benjamin.duplex@cea.fr

<sup>(2)</sup> CEA, DEN  
DER/SSTH/LDAS  
17, rue des Martyrs  
Grenoble  
F-38054, France

fabien.perdu@cea.fr

<sup>(3)</sup> LSIS UMR CNRS 6168  
ESIL Case 925  
163 Avenue de Luminy  
Marseille cedex 9  
F-13288, France

benjamin.duplex@lsis.org  
gilles.gesquiere@lsis.org  
marc.daniel@lsis.org

## ABSTRACT

Physical codes are dependent on a discipline and describe the reality through equations solved on a geometry. Strong links, or couplings, between them are set up to increase the result accuracy, leading to data exchanges. Mechanical codes calculate geometrical deformations which must be transferred by couplings. Their direct exchange is generally impossible if coupled codes compute with different models of the same geometry (in practice different meshes). Moreover, two meshes can only share border(s). This paper focuses on this case: one mesh must follow the border deformation of the mechanical code mesh.

## Keywords

radial basis function, CS-RBF, physical code coupling, deformation exchange, experimental device

## 1 INTRODUCTION

The French Atomic Energy and Energy Alternatives Commission <sup>1</sup> (CEA) uses codes to simulate physical behaviours in nuclear reactors. They are specialized to a specific domain, such as mechanics, thermohydraulics or fuel behaviour. The current CEA approach is to couple codes to improve the final result accuracy, leading to data exchanges between codes. These exchanges can be of various forms: point values (*e.g.* a temperature at a point), data fields (*e.g.* a pressure on a surface) or geometrical deformations. A mechanical code calculates geometrical modifications. Previous works have been made on the data exchange problem

([BDS07,PV08]). In this paper, we focus on the transfer of deformations calculated by a mechanical code to a second one in the particular but very important case where code geometries only share border(s).

The code geometrical models could be CSG or meshes. In this work, only meshes are considered because it corresponds to the most encountered situation. Each code is based on a particular mesh to optimize its computations with specific dimension (*e.g.* surfacic or volumic), properties (*e.g.* regularity and uniformity) and cell types (*e.g.* triangles, quadrangles). Most of them are composed of hundreds of thousand cells up to millions of cells. It must be added that the mesh deformations must keep the computed data attached to the mesh for further applications, which is against a full remeshing.

Based on the differences between meshes, it appears useful to define a continuous function to model the geometrical deformations. This function can easily be sampled to any point where a displacement must be computed. One of the constraint is to interpolate the deformations and not to approximate them. Another and natural constraint is that the method must have a lower processing time order than the code calculation: *e.g.* hours when the code execution time is in days (involving generally several hundred thousand cells).

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, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

<sup>1</sup> CEA: French Atomic Energy and Energy Alternatives Commission ([http://www.cea.fr/english\\_portal](http://www.cea.fr/english_portal))

The remainder of the paper is structured as follows: section 2 reviews related work. Section 3 presents our method and its results are proposed in section 4.

## 2 RELATED WORK

The problem is to transfer an interpolated deformation between meshes that only share border(s).

In [DGDP10], we addressed the problem of transferring the deformations between meshes modelling the same geometry. We introduced a function  $F_\alpha$  computed on a network of radial basis functions (RBFs) after a mandatory simplification step to receive reasonable computing times. The RBF network is a space deformation function ([Wen05]), the deformed area being the same that the one where the network is computed. This approach is thus not directly applicable.

To address our specific problem, the methods must be decomposed into two steps: the deformation of the first area must be applied to the border of the second area and then the inner part of the second area must be processed subject to its border deformation. For the second step, different approaches exist. Full remeshing techniques ([FG08]) are possible. However, it involves the loss of the data attached to the former inner nodes. In [CSW09], the authors apply a mass-spring model. It changes the inside point position iteratively until the complete stabilization. This method is unfortunately too expensive in our case. In [YK09], a smoothing method optimizes the aspect ratio and reduces the element skew of cells. However, some nodes initially placed inside could finally be outside the mesh to smooth if new borders are within the initial mesh. In [Hel03], a Laplacian method is employed. Results shown are good but deteriorated for large displacements. RBFs are used in [dBvdSB07]. They interpolate the variation of node positions located on the mesh border. RBFs involve to solve a linear system whose size depends on the number of nodes to interpolate. In our case, this number is very high. It induces a computational time that exceeds target set by the CEA. It is the reason why we introduced function  $F_\alpha$  ([DGDP10]).

## 3 THE PROPOSED METHOD

### 3.1 Overview of function $F_\alpha$

$F_\alpha$  combines a simplification method (not detailed in this paper but based on [GH97]) which keeps  $\alpha$  nodes, and a RBF network which interpolates these points. Due to the simplification step,  $F_\alpha$  does not interpolate the exact deformation. As the kept nodes are considered the most important ones according to the simplification method criterion, the induced error is very low. These nodes are called landmark points  $\{L_i\}_{i=1,\dots,\alpha} \in \mathbb{R}^3$ . For any point  $P \in \mathbb{R}^3$ , function  $F_\alpha : \mathbb{R}^3 \rightarrow \mathbb{R}^3$  is defined by:

$$F_\alpha(P) = \sum_{i=1}^{\alpha} \lambda_i \cdot \phi(\|P - L_i\|_2) + \mathcal{L}(P)$$

where  $\phi$  is the  $C^2$  Thin-Plate Spline (TPS) such that  $\phi(r) = r^4 \cdot \log(r)$ . They are centered on landmarks  $L_i$ .  $\lambda_i \in \mathbb{R}^3$  are the network coefficients and  $\mathcal{L}$  is an affine function corresponding to global displacements:

$$\mathcal{L}(P) = \sum_{c=1}^3 \lambda_{\alpha+c} \cdot P^{(c)} + \lambda_{\alpha+4}$$

where  $P^{(c)}$  is the  $c^{\text{th}}$  coordinate of point  $P$ . With the same notations, the additional constraints are imposed to take into account the polynomial terms of  $\mathcal{L}$ :

$$\sum_{i=1}^{\alpha} \lambda_i \cdot L_i^{(c)} = \sum_{i=1}^{\alpha} \lambda_i = 0, \quad \forall c = 1, 2, 3$$

Centers  $\{L_i\}_{i=1,\dots,\alpha}$  of TPSs are known value positions that function  $F_\alpha$  must interpolate. If  $L_i$  (resp.  $L_i'$ ) denotes the  $i^{\text{th}}$  node of initial (resp. deformed) mesh used by the mechanical code, the interpolation condition is:  $F_\alpha(L_i) = L_i'$ ;  $i = 1, \dots, \alpha$ . The network coefficients are thus calculated by solving a linear system involved by these equations. We define matrix  $\Psi \in \mathbb{R}^{\alpha \times \alpha}$  by:

$$\Psi = \left( \phi(\|L_i - L_j\|_2) \right)_{1 \leq i, j \leq \alpha}$$

and matrix  $\xi$  by  $\xi = (x_i, y_i, z_i, 1)_{1 \leq i \leq \alpha} \in \mathbb{R}^{4 \times \alpha}$ , with  $(x_i, y_i, z_i) \in \mathbb{R}^3$  the coordinates of point  $L_i$ . Matrix  $A \in \mathbb{R}^{(\alpha+4) \times (\alpha+4)}$  of the linear system induced by the RBF network is defined by:

$$A = \begin{pmatrix} \Psi & \xi \\ \xi^T & \mathbf{0} \end{pmatrix}; \quad \text{with } \mathbf{0} = (0)_{4,4} \in \mathbb{R}^{4 \times 4}$$

The RBF calculation using TPSs implies a dense matrix  $A$ , symmetric but with a zero-filled diagonal. Parameter  $\alpha$  must be small enough to have acceptable computation time and memory space used, involving an error during the deformation calculation.

The disadvantages exposed above are due to two aspects of TPSs. Functions  $\phi(r)$  are exponentially growing, global to the whole data and equal to zero when  $r = 0$ .  $\Psi$  is symmetric, because:  $\forall 1 \leq i, j \leq \alpha$ ,  $\phi(\|L_i - L_j\|_2) = \phi(\|L_j - L_i\|_2)$ , but not positive definite since  $\phi(\|L_i - L_i\|_2) = \phi(0) = 0$ .  $A$  is dense and symmetric with zero diagonal. That prevents to use efficient methods for solving the linear system, like iterative methods.

### 3.2 Definition of a new function $F_\alpha^\sigma$

In order to overcome the difficulties detailed in the previous section, we propose in this paper to use functions with compact support. Their advantage is to vanish when the distance between two centers  $L_i$  and  $L_j$  of the RBF exceeds a threshold, leading to sparse matrices  $\Psi$  and  $A$ . According to [RA10], the most efficient functions are the CS-RBF, Compactly-Supported Radial

Basis Function, proposed by Wendland ([Wen05]). For meshes in  $\mathbb{R}^3$ ,  $\phi(r) = (1-r)^4 \cdot (4r+1)$  if  $r < 1$ ,  $\phi(r) = 0$  otherwise. This function is defined on interval  $[0, 1]$  with  $\phi(0) = 1$  and  $\forall r \in ]0, 1[$ ,  $0 < \phi(r) < 1$ . This provides a matrix  $\Psi$  symmetric positive definite, sparse and diagonally dominant. However, this function requires a support size. Although works have already been focused on solving this problem ([OBS05]), it remains open.

We introduce parameter  $\sigma$ . For a given CS-RBF, we calculate the minimal radius  $s_i$  of the ball containing  $\sigma$  centers  $\{L\}$ . The global support size  $S$  is obtained to ensure that every CS-RBF involve at least  $\sigma$  centers (e.g.  $S = \max_{1 \leq i \leq \alpha} s_i$ ). Its calculation is in  $O(\alpha \cdot \log(\alpha))$  and is only done once. A new interpolation function, denoted  $F_\alpha^\sigma$ , is defined. It corresponds to  $F_\alpha$  ([DGDPI0]) with TPS replaced by CS-RBF. Its support size is calculated via  $\sigma$  as detailed above.

### 3.3 Mesh deformation

Transferring deformations to a mesh only sharing borders with the one used by the mechanical code can be realized through  $F_\alpha^\sigma$ . Our whole method is detailed in figure 1, every step being described below. To begin, a mesh is computed from the geometry of the whole device (1a). The mechanical code calculation

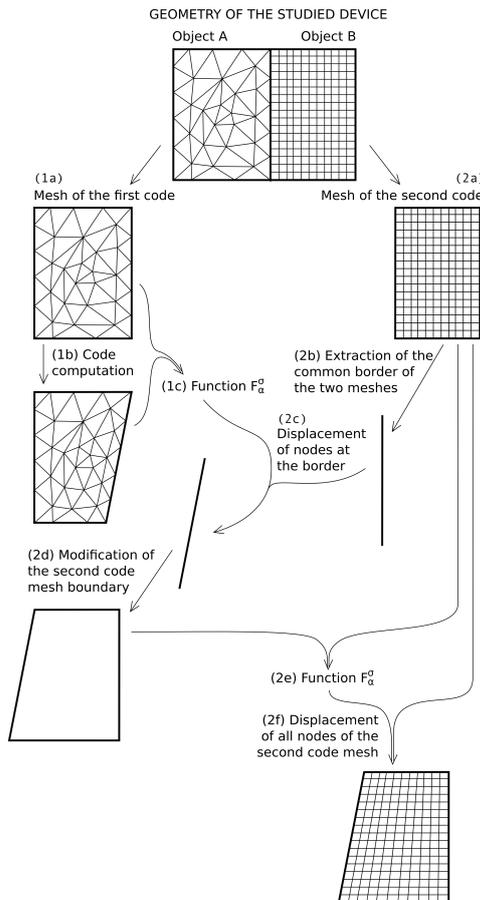


Figure 1: The outlines of our method

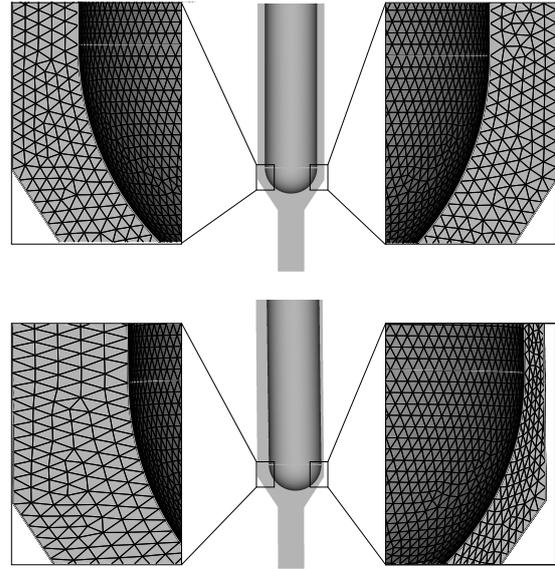


Figure 2: The thermohydraulic code mesh deformation (top: original, bottom: modified by our method)

is launched (1b). A function  $F_\alpha^\sigma$  is computed after this step to interpolate the geometrical deformations computed by the mechanical code (1c). The initial mesh of the second code is then computed (2a), again from the geometry of the whole device. To be able to modify this mesh, its borders are extracted (2b). Common parts between these borders and the first code mesh are extracted and displaced (2c) applying function  $F_\alpha^\sigma$  previously obtained. These modifications can then be applied to all borders of the second code mesh (2d). The boundary of the second code mesh have been displaced. A function  $F_\alpha^\sigma$  is finally calculated to interpolate the evolution of the second mesh border position (2e). Once applied to all inner nodes, the second mesh is completely modified (2f).

## 4 EXPERIMENTAL RESULTS

The use case is an experimental device composed by two pipes placed inside a research reactor. The first one, called pressure tube, encapsulates an experiment and isolates it from the rest of the reactor. It is positioned into a second one, called a device holder. Between these pipes, a water flow evacuates the heat emitted by the experiment. The case presented is the thermohydraulic study of the water.

The reactor core heats both pipes leading to anisotropic tube dilatations and water thickness variation. Cooling is no longer homogeneous inducing different pipe deformations and some thermohydraulic phenomena can occur. A coupling between a mechanical code and a thermohydraulic code is valuable. The presented method allows to modify the thermohydraulic code mesh according to the geometrical modifications calculated by the mechanical code. Both initial meshes do not discretise the same geometry. The

first one is a mesh of both tubes, whereas the other represent the water separating them. Both meshes share common borders. The first mesh is composed of 14,110 nodes and 37,330 tetrahedra, and the second one 223,737 nodes and 1,025,622 tetrahedra. Figure 2 illustrates our results. Both tubes are deformed and the water section is then modified. The most important area is located at the bottom of the pressure tube. Pipe deformations involves a water section variation: a shrinkage on one side and an enlargement on the other side. One can see on figure 2 that our method allows to shrink and to stretch smoothly the mesh cells. Computing time of steps is detailed in table 1. The device holder deformation is interpolated by function  $F_{\infty}^{100}$ , noted  $F_{(DH)}$ , where  $\alpha = \infty$  means the simplification is not processed. The same choice, noted  $F_{(PT)}$ , is made for the pressure tube.  $F_{(W)}$  corresponds to the water mesh modification. Its parameters, chosen experimentally, are  $\alpha = 5000$  and  $\sigma = 100$ . The whole use case calculation is achieved in 13min 19s. In comparison, it is performed in 1h 7min 37s with function  $F_{\alpha}$ . The computer used is a PC 64bits under Linux, using a single core at 1.6GHz and with 4GB of RAM.

	step	computing time	# nodes
$F_{(DH)}$	calculation	2min 04s	9,302
	application	3min 24s	52,453
$F_{(PT)}$	calculation	36s	4,816
	application	1min 22 s	39,962
$F_{(W)}$	calculation	37s	5,000
	application	5min 07 s	131,322

Table 1: Computing time detail for each step

Table 1 shows the method performances in terms of computation time compared with the number of nodes used. For information, the thermohydraulic code requires several days while our method updates its mesh within less than 15min. As the experimental device study is not yet achieved, we cannot definitely conclude about our method accuracy, but thermohydraulic experts at the CEA validate our results.

## 5 CONCLUSION AND DISCUSSION

The proposed method allows to transfer deformations computed on a mesh to another mesh sharing common borders. Based on previous works, a new function  $F_{\alpha}^{\sigma}$  is established. It interpolates the displacement of a large number of nodes. This function is the result of a simplification step and the computation of a network of Radial Basis Functions having a compact support. The constraints imposed by the CEA are all well respected. The complete validation of the method will be performed when the device development will be achieved, but according to specialists, results are promising.

An interesting perspective would be to calculate automatically  $\sigma$ , parameter controlling the compact support. For the non-uniform mesh case, one idea might be to establish a grid centered in each RBF center and calculate the value of  $\sigma$  to have all grid cells which contain at least one (or more) other(s) center(s).

## ACKNOWLEDGEMENTS

This research is supported by the French Atomic Energy and Energy Alternatives Commission (CEA).

## REFERENCES

- [BDS07] BONACCORSI T., DANIEL M., SALVO J. D.: Data exchange interface and model for coupling softwares in nuclear reactor simulations. In *Inter. conf. in PLM* (Italy, 2007), pp. 207–216.
- [CSW09] CUI T., SONG A., WU J.: Simulation of a mass-spring model for global deformation. *Frontiers of EEE in China* 4 (2009), 78–82.
- [dBvdSB07] DE BOER A., VAN DER SCHOOT M., BIJL H.: Mesh deformation based on radial basis function interpolation. *Computers & Structures* 85, 11-14 (2007), 784 – 795.
- [DGDP10] DUPLEX B., GESQUIÈRE G., DANIEL M., PERDU F.: Transfer of mesh deformations between physical codes. In *Curves and Surfaces conference* (june 2010).
- [FG08] FREY P. J., GEORGE P. L.: *Mesh Generation. Application to Finite Elements*. ISTE, 2008.
- [GH97] GARLAND M., HECKBERT P. S.: Surface simplification using quadric error metrics. In *SIG-GRAPH* (USA, 1997), pp. 209–216.
- [Hel03] HELENBROOK B.: Mesh deformation using the biharmonic operator. *Inter. journal for numerical methods in eng.* 56, 7 (2003), 1007–1021.
- [OBS05] OHTAKE Y., BELYAEV A., SEIDEL H.-P.: 3d scattered data interpolation and approximation with multilevel compactly supported rbfs. *Graphical Models* 67, 3 (2005), 150 – 165.
- [PV08] PERDU F., VANDROUX S.: System / CFD Coupling for Reactor Transient Analysis. An Application to the Gas Fast Reactor with CATHARE and TRIO\_U. *ICANPP* (june 2008).
- [RA10] RENDALL T., ALLEN C.: Reduced surface point selection options for efficient mesh deformation using radial basis functions. *Journal of Computational Physics* 229 (2010), 2810 – 2820.
- [Wen05] WENDLAND H.: *Scattered Data Approximation*. Cambridge University Press, 2005.
- [YK09] YILMAZ A., KUZUOGLU M.: A particle swarm optimization approach for hexahedral mesh smoothing. *International Journal for Numerical Methods in Fluids* 60, 1 (2009), 55–78.

# POSTER: Parametric 3D model of sperm cell surface applied in confocal microscopy

Jozef Marek  
Inst. Experimental  
Physics SAS,  
Watsonova 47,  
Kosice, Slovakia  
marek@saske.sk

Ivana Uhrinova  
Dept. Biology,  
P.J. Safarik Univ.  
Moyzesova 11  
Kosice, Slovakia

Erna Demjen  
Inst. Experimental  
Physics SAS,  
Watsonova 47,  
Kosice, Slovakia  
demjen@saske.sk

Zoltan Tomori  
Inst. Experimental  
Physics SAS,  
Watsonova 47,  
Kosice, Slovakia  
tomori@saske.sk

## ABSTRACT

Confocal microscopy combined with 3D graphics enable to study spatial characteristics of biological objects on microscopic level. Human sperm cells have a typical shape with marked asymmetry in its frontal-back part. 2D analysis with optical microscope gives mainly information about horizontal plane of the cell. Presented parametrical model provides important geometrical parameters describing 3D shape of the cell surface using 3D reconstruction of horizontal cross-sections acquired by a confocal microscope.

## Keywords

3-D Modeling, Surface Reconstruction, Parametric Model, Confocal Microscopy, Sperm Cell.

## 1. INTRODUCTION

The relationship between the shape and the function of a cell is the subject of intensive research. Moreover, cell morphology represented by texture and shape indexes is applicable in medicine ([TDH+08]). Sperm cell (especially its head) has a typical shape with marked asymmetry of its frontal-back part. Although there were attempts to describe the 2D contour of the cell's head [BCV05], 3D models are still rare. The reason is that traditional optical microscopy cannot provide high-resolution 3D images. Atomic force microscopy (AFM) [KCS+05] and confocal laser scanning microscopy (CLSM) [DBF+97] opened new possibilities to obtain detailed information about 3D topology of sperm cell. Most of images show a marked flatness of the frontal part of the cell. This deformation (hydrodynamic profile) is not included in 2D models and can be an important factor of the cells movement in the surrounding environment.

The output of confocal microscope is a series of

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.

digital images representing individual cross-sections acquired from the different depth of the sample. 3D surface can be reconstructed from these optical sections by using image processing techniques. 2D contours of sperm cell at all levels of depth consist of pixels that can represent surface voxels in 3D. Fitting of all these voxels by a proper parametric model allows both efficient visualization and shape analysis. A novel parametric model of sperm cell surface is proposed in this contribution. The shape analysis allows elimination of artifacts in fluorescent microscopy where their irregular shape and random position lead to the variable intensity of emitted light. We expect that individual parameters can serve as a feature vector for classification of sperm cells.

## 2. MODEL

A lot of surface models in computer graphics are based on superquadrics however, they are not proper to fit wrinkles in the front part of sperm cell. The basis of our model (Figure 1) is an ellipsoid with main axes  $a_0$ ,  $b_0$ ,  $c_0$  (in y, z, x directions) however, its surface is modified in cylindrical coordinate system through a parametrical function  $\rho(v, \theta)$  with variables  $\theta$  (lateral angle) and  $v = t/c_0$  (relative distance) :

$$\rho(v, \theta) = \sqrt{\frac{1 - v^2}{(\cos\theta/a(v))^2 + (\sin\theta/b(v))^2}} \quad (1)$$

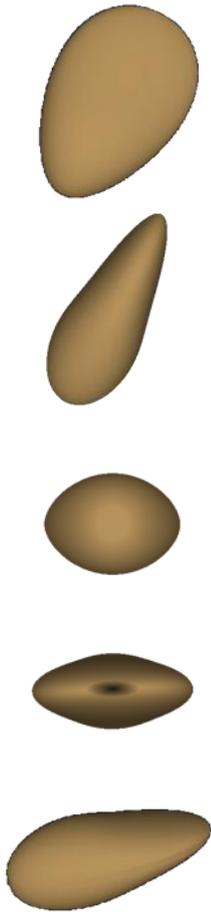
where for the main axes holds

$$a(v) = a_0 (1 + \delta_a v) \quad (2)$$

$$b(v) = b_0 \cdot flat \quad (3)$$

and flatness is given by the equation

$$flat = 1 - d_f / (1 + \exp[-k_f(v - v_f)]) \quad (4)$$



**Figure 1.** Illustration of a 3D-model surface of the cell (top, side, back, frontal and a rotated view), depicted using K3DSurf [K3DS].

The above defined model has 7 parameters specified in Table 1, Figure 3 and incorporates basic shape features of the sperm cell (Figure 2):

After backward transformation of Eq.(1)-(4) to the Cartesian coordinate system:

$$x(v, \theta) = v \quad (5)$$

$$y(v, \theta) = \rho(v, \theta) \cos(\theta) \quad (6)$$

$$z(v, \theta) = \rho(v, \theta) \sin(\theta) \quad (7)$$



**Figure 2.** Selection of horizontal slices of a real cell acquired by confocal microscope (inverted).

### 3. CONFOCAL MICROSCOPY

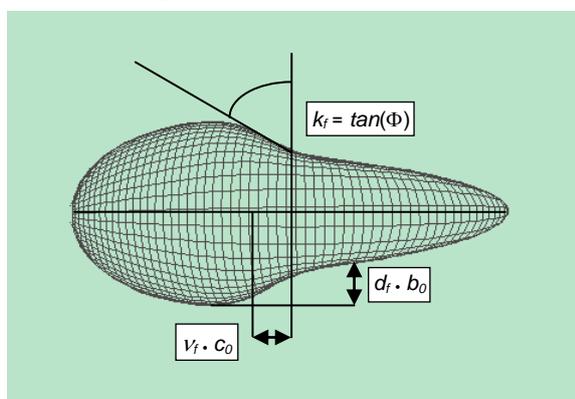
Purified sperm cells were obtained from the patients of Centre for Assisted Reproduction with the institutional ethical approval. After fixation, human sperm cells were stained with a fluorescent dye Ethidium Bromide. Stained cells were then placed into polyacrylamide gel environment to guarantee

their stable position during scanning. Subsequently, the microscopic preparation was created by the standard way.

$a_0, b_0, c_0$	the main axes of the ellipsoid
$\delta_a$	asymmetry of the cell's contour in its horizontal plane
$v_f$	location of the vertical flatness of the cell relative to its center
$k_f$	steepness of the vertical flatness
$d_f$	depth of the vertical flatness

**Table 1. Parameters used in shape model**

Samples were scanned by fluorescence confocal microscope which offers a series of digital images captured in different depth of sample (0.4 micrometers apart).



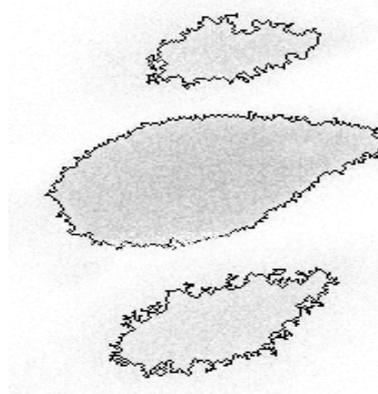
**Figure 3. Model parameters interpretation, depicted using K3DSurf [K3DS].**

Two types of confocal microscopes have been used - Leica DM 2500 CSQ V-VIS and Leica TCS SP5 (objective 100, resolution xy-139nm, z-236nm) both supplied with Leica Application Suite Advanced Fluorescence (LAS AF). Surface points coordinates  $x, y, z$  were calculated from contours acquired by segmentation of calibrated microscopy scans of horizontal cross-section (Figure 2). A simple thresholding segmentation algorithm followed by edge tracking contour detection was used (Figure 4). Approximation of the cell surface with the defined model resulted in noise reduction.

#### 4. DATA PROCESSING

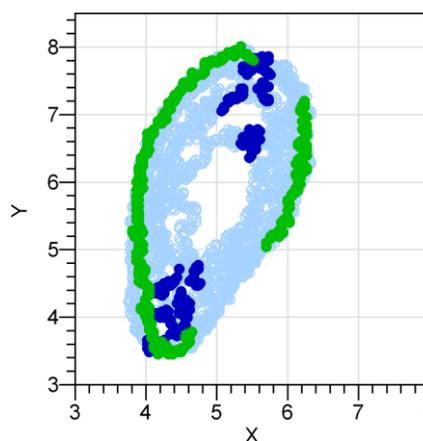
As the cell's centre and orientation differs from the basic position of the model, it is necessary to use 6 additional parameters (3 coordinates of the center and 3 angles of the orientation) for model analysis. In

order to increase the reliability of calculations and accelerate computing, parameters of the cell's placements and orientation are calculated at first by sequential fitting of the surface with the main planes of the ellipsoid. Their intersection defines the main axis and its center the cells' center (Figure 5).



**Figure 4. Segmentation of individual optical sections.**

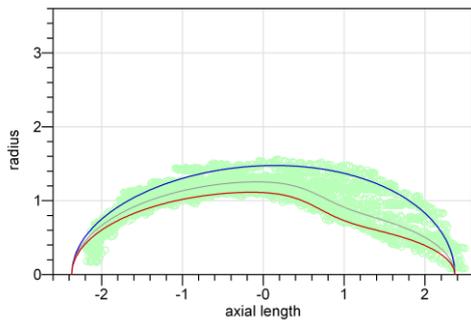
Consecutively, the cell surface points are transformed into cylindrical coordinate system with origin in the cell's centre oriented toward its main axis and these data are used in the subsequent analysis. Downhill simplex algorithm was used for least square optimization of proposed parametric model.



**Figure 5. Schematic drawing of surface points projection (light blue dots) and projection of the nearest points to the cell main planes fits (blue – horizontal plane, green– vertical plane) from side view.**

## 5. RESULTS

Nonlinear regression methods (Figure 6) were used to gain geometrical parameters of a real cell:  $a_0 = 1.5 \mu\text{m}$ ,  $b_0 = 1.1 \mu\text{m}$ ,  $c_0 = 2.4 \mu\text{m}$ ,  $\delta_a = 0.06$ ,  $t_f = v_f \cdot c_0 = 0.7 \mu\text{m}$ ,  $k_f = 11.5$ ,  $d_f = 0.35$  with standard deviation of the surface  $0.1 \mu\text{m}$  and with correlation coefficient 0.94. Figure 7 shows the visualized model of sperm cell surface fitting the real image data acquired by the confocal microscope.



**Figure 6. Longitudinal half-contours of the model surface of the cell with the real cell contours in the background. Three longitudinal contours corresponding to angles 0°-blue, 45°-gray, 90°-red are depicted.**

## 6. CONCLUSION

We have presented 3D-reconstruction of real human sperm cell surface from confocal microscopy optical sections using novel parametric surface model. The extracted parameters occur in region of real cell dimensions and as we believe could be useful to 3D-shape cell classification.

## 7. ACKNOWLEDGMENTS

This work was supported by the Slovak scientific grant agencies APVV (0682/07, SK-CZ 0137/09), VEGA (2/0164/08), by CEX NANOFLUID and by the European Regional Development Fund (Project 26220120033). The help of Dr. Barbara Radochova (Physiol. Institute ASCR) is appreciated.

## 8. REFERENCES

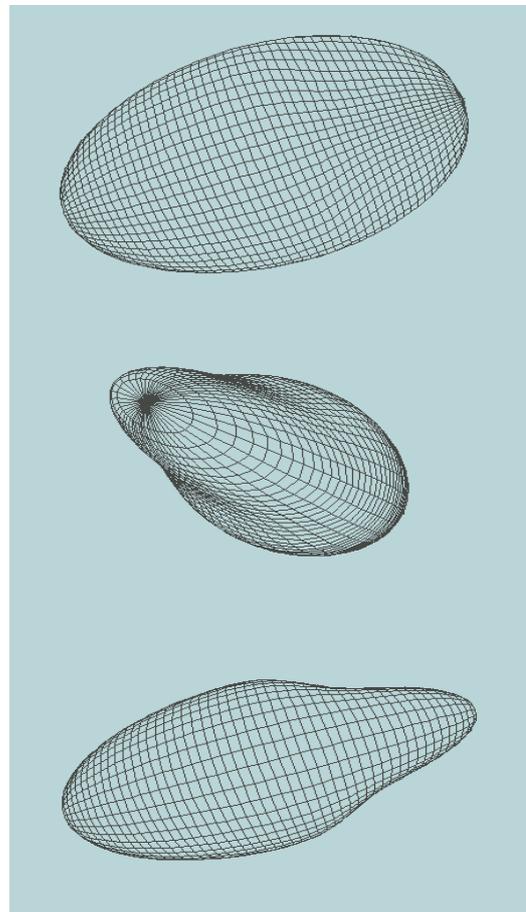
[BCV05] Beletti, M.E., Costa L.F. and Viana, M.P. A spectral framework for sperm shape characterization. *Computers in Biology and Medicine* 35, pp. 463-473, 2005.

[DBF<sup>+</sup>97] Diaspro, A., Beltrame, F., Fato, M., Palmeri, A., and Ramoino, P. Studies on the structure of sperm heads of *Eledone cirrhosa* by means of CLSM linked to bioimage-oriented devices. *Microscopy Research and Technique* 36, pp.159-164, 1997.

[K3DS] <http://k3dsurf.sourceforge.net/>.

[KCS<sup>+</sup>05] Kumar, S., Chaudhury, K., Sen, P. and Guha, S.K. Atomic force microscopy: a powerful tool for high-resolution imaging of spermatozoa. *Journal of Nanobiotechnology* 3, 1, 2005.

[TDH<sup>+</sup>08] Thibault, G., Devic, C., Horn, J.F., Fertil, B., Sequeira, J. and Mari, J.L. Classification of cell nuclei using shape and texture indexes. In *Conf. Proc. WSCG 2008Poster papers*, pp. 25-28, 2008



**Figure 7. Visualization of the surface model of sperm cell head (top, rotated and side views), depicted using K3DSurf [K3DS].**

# An exact hierarchical geometric model. Combining remeshing and spatial decomposition

A. Martinez<sup>1</sup>, J. Jimenez<sup>1</sup>, F. Paulano<sup>1</sup>, R. Pulido<sup>1</sup> and F. Feito<sup>1</sup>

<sup>1</sup>Departamento de Informatica. Universidad de Jaen

## ABSTRACT

The use of hierarchical spatial decomposition in 3D scenes in order to manage the complexity of the objects is a well known approach. The main problem with this technique is the updating process when the mesh is modified. Any deformation or rotation means a new complete reconstruction of the structure. By other way remeshing techniques modify the structure of a mesh in order to achieve a given quality requirement. In this study a combination of remeshing techniques and hierarchical spatial decomposition is presented. Our goal is to develop an new model applying a remeshing process based on the hierarchical structure elements. This new model allows to extend one deformation in the spatial decomposition to the mesh. The tetra-tree is chosen as the spatial decomposition because of its advantages in relation to the remeshing algorithm. Tests with medium meshes with the new model were performed with good results.

**Keywords:** Normal orientation, mesh repair, visibility, patch connectivity, CAD tools.

## 1 INTRODUCTION

Nowadays the visualization of complex scenes is common in interactive environments. The complexity of the objects inside the 3D interactive-scene and the time requirements (real-time) force us to develop simplification techniques to fulfill the requirements without decreasing the detail. Spatial decomposition is one of these approaches that solves this problem [15]. One advantage of these structures is that can be implemented hierarchically, so its complexity can be adapted to different environments. The main disadvantage is the updating process that is forced by any modification (deformation) of the original mesh. This problem appears because the relation between triangles and nodes of the spatial decomposition is not unique, so one triangle could belong to more than one node.

On the other hand remeshing techniques are common in many areas of computer graphics areas such as: surface sampling [3], surface parametrization [14], remeshing irregular geometry [1], improving mesh quality [16] and mesh approximation [6]. These techniques modify the triangles of the mesh in order to fulfill some quality requirement where the complexity is often constrained. In our paper a combination between remeshing techniques and spatial decomposition is proposed. We call *exact model* if each triangle of the mesh belongs to only one node, on otherwise is an

*non-exact model*. Our goal is adapting the mesh to the nodes of the spatial decomposition in order to develop an exact geometric modeling. Previous work [5, 10, 13] uses spatial decomposition to define the way to generate new meshes. This remeshing transforms the triangles mesh inside the nodes into new nodes of the structure. In this paper a novel hierarchical spatial decomposition based on the work by Jimenez [7] is proposed, but with some modifications in order to overcome the limitations of the classical model and for developing an exact geometric model.

## 2 REMESHING

In many applications, a remeshing procedure is necessary to increase the level of detail, generate non-homogeneous triangulation or improve triangulated meshes [5, 10, 13]. Besides needing to reduce the complexity of the meshes, the mesh quality must be improved. Other remeshing techniques focus on compatible refinement, hierarchical simplicial trees and the definition of the maximum factor of mesh growth [4]. Alliez et al. [2] performed a complete survey of remeshing techniques, defining remeshing as: "*Given a 3D mesh, compute another mesh, whose elements satisfy some quality requirements, while approximating the input acceptably*". In this paper, a remeshing procedure is applied to adjust the original mesh to a hierarchical spatial decomposition, so our quality requirements are based on the nodes of the hierarchical spatial decomposition. The remeshing is localized because the procedure is not applied to the whole mesh but only to a subset of triangles which do not belong completely to a node of the hierarchical decomposition. This remeshing procedure could be considered as a high quality remeshing, taking into consideration the fact that the mesh is subdivided in

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.

Plzen, Czech Republic.  
Copyright UNION Agency – Science Press

order to obtain a new mesh with more triangles than the original one. If in previous applications this approach is focused on adapting the mesh to some properties of the mesh such as curvature [1], in our work the mesh is adapted to the hierarchical spatial decomposition.

### 3 COMBINING HIERARCHICAL SPATIAL STRUCTURES AND REMESHING TECHNIQUES

The nodes of the tetra-tree are tetra-cones that are tetrahedrons with their base at infinity. So the problem of intersecting nodes and shared triangles is reduced to a triple intersection triangle and lateral tetra-cone face, that is a triangle - triangle 3D intersection. The triangle - triangle intersection is easily reduced to a segmented/ray - triangle intersection where each ray is an edge of the triangle and the triangle is one of the lateral faces of the tetra-cone. The ray - triangle intersection is an important problem in many computer graphic areas. The classical algorithm used for the ray - triangle intersection is the one proposed by Möller [11]. The main problem of this approach is how it deals with limit cases which forces us to trace new rays. A review of ray-triangle intersection algorithms is shown in [9]. To overcome the problem of the limit cases we use a robust point in polygon test based on barycentric coordinates [8] and the intersection algorithm proposed by the same author [9] where the limit cases are found directly during the study of the value of barycentric coordinates. In this scheme the use of tetra-tree is a major advantage because, the classification method for the hierarchical structure, the point in polygon test and the ray-intersection algorithm are all based on barycentric coordinates so some calculations may be reused. After the combined inclusion and intersection procedure, all the possible cases of triangle - triangle intersection are studied and one of the splitting patterns is chosen (see figure 2). The splitting patterns are selected to be as simple and efficient as possible, attempting to reduce the number of triangles generated and being invariant to the edges intersected. In this paper this scheme has been applied in a general form without any precalculation during the building of the spatial structure, so this approach could be applied to any spatial structure that is composed or reduced using tetrahedrons or tetra-cones.

The special case are divided into rejected special cases, which are directly removed from the remeshing procedure, and degenerated special cases where the remeshing is applied but a test to remove degenerated triangles is performed as well. In general geometric problems the special cases are not usual or are very rare [12], but in our case a previous splitting process often generates many special cases. If these cases are not explicitly controlled the splitting process could have no ending. The Jimenez algorithm deals with the limit cases naturally using the barycentric coordinates, so the

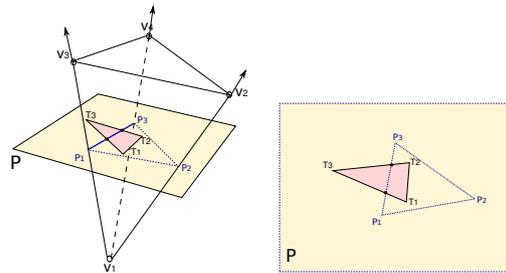


Figure 1: The triangle - tetra-cone intersection can be easily expressed as a triangle-triangle intersection using the plane that contains the triangle.

special cases are handled directly with no additional operations. If the mesh is required to be compatible some post-processing algorithm could be applied [4].

#### 3.1 Cases of intersection between two triangles and subdivision patterns

There are 11 cases of intersection between two triangles (see figure 2). In these images we consider the blue triangle as the triangle of the mesh to split and the red one as the triangle of the intersection of the plane that contains the triangle and the tetra-cone (see figure 1). Taking into consideration the assumption that the tetra-cone is larger than the triangle to be classified, the most probable subdivision cases are (sorted by decreasing probability): case 3, case 4, case 1, case 8, case 7 and case 11. Cases 2, 5, 6 and 9 are plausible theoretically but did not appear on our tests [see section 4]. So most of the triangles intersect only one face of the tetra-cones and have two intersection points. The tests performed confirm this probability distribution. Once the intersection case is defined, a subdivision pattern is applied. These patterns have been defined attempting to reduce the number of degenerated triangles. An additional test to reject degenerated triangles is also used.

## 4 EXPERIMENTAL RESULTS

The new model has been tested over medium-size meshes, similar to those most usually used in computer graphics and common applications. The first mesh has 40000 triangles and 20002 vertices. Table (4) shows how the size of the mesh increases according to tetra-tree levels. The amount of new primitives increases geometrically in relation to the number of tetra-cones which increases exponentially. The distribution of the cases where the subdivision is applied is shown in table (4) too.

The mesh before the remeshing procedure and after is shown on figure [5], and the green triangles are the shared triangles found on the mesh.

More tests are summarized on tables 6, 7 and 8.

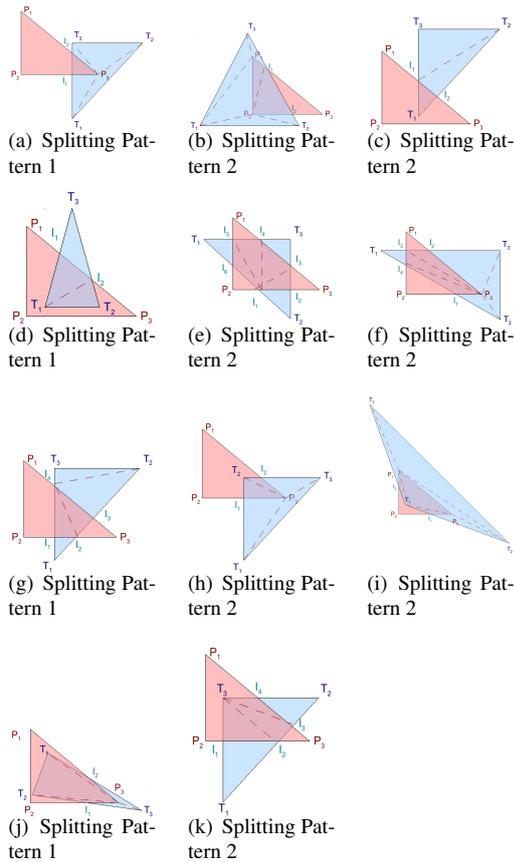


Figure 2: Intersection cases and splitting patterns.

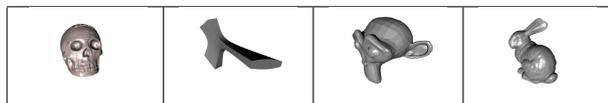


Figure 3: Medium-size mesh used.

TT Depth	TT Generated	Final triangles	Final points
1	24	47232	27334
2	120	53599	33610
3	504	64967	44978
4	2040	82757	62768

Figure 4: Evolution of the number of triangles and points when subdivision is applied. Distribution of the intersection cases. C is the intersection case number, and TTD is the tetra-tree depth.

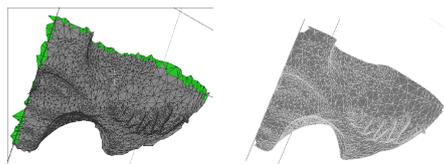


Figure 5: Before (left) and after (right) the remeshing algorithm, the shared triangles have been removed. In green, the shared triangles from the original mesh

TT Depth	Final triangles	Final points
1	2648	1954
2	3533	2834
3	4413	3707
4	4803	4097

Figure 6: Mesh, new size of the mesh and cases. The original mesh has 1854 triangles and 1172 points. TTD is the tetra-tree depth

TT Depth	Final triangles	Final points
1	3131	2072
2	4044	2978
3	4634	3566
4	5462	4389

Figure 7: Mesh, new size of the mesh and cases. The original mesh has 2180 triangles and 1132 points. TTD is the tetra-tree depth

TT Depth	Final triangles	Final points
1	74173	40667
2	84008	50501
3	93907	60403
4	127036	93565

Figure 8: Mesh, new size of the mesh and cases. The original mesh has 69459 triangles and 35947 points. TTD is the tetra-tree depth

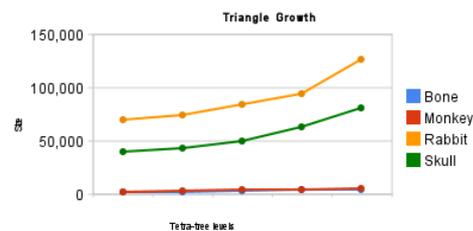


Figure 9: Evolution of the mesh size after the remeshing procedure in relation with the tetra-tree level.

## 5 CONCLUSIONS AND FUTURE RESEARCH

A new hierarchical spatial decomposition for dealing with complex objects has been presented. This new approach is based on regular spatial decomposition performed by tetra-trees. In order to achieve this goal, a local remeshing method is applied. This method transforms the relation between triangle and tetra-cone into a direct relationship, increasing the size of the mesh. The splitting process adds triangles to the original mesh, but this mesh is simplified with the hierarchical structure. So we increase the time for tetra-tree building (only performed at the beginning), attempting to avoid an updating process on the interaction environment.

Various tests with real meshes were performed to compare the original meshes with the remeshed ones. Regarding the main disadvantage of our model, the computational cost of the building process, some preliminary work for translating the building process to the GPU are in progress. The geometric operations of the split method are easily implemented in GPU as well so the whole process could be performed in the GPU thus

reducing the associated computational cost. With regard to the geometrical operations applied to hierarchical decomposition some ideas are under development, such as: a fast way to locate nodes and determine the concavities and holes in the mesh, definition of depth levels to classify the triangles inside the nodes, and a space-filling index to perform transversal operations on the tree.

## 6 ACKNOWLEDGEMENTS

This work has been partially supported by the Spanish Ministry of Education and Science and the European Union (via ERDF funds) through the research project TIN2007-67474-C03-03, by the Consejería de Innovación, Ciencia y Empresa of the Junta de Andalucía through the research projects P06-TIC-01403 and P07-TIC-02773, and by the University of Jaén through the research project UJA-08-16-02, sponsored by Caja Rural de Jaén. The rabbit mesh comes from The Stanford 3D Scanning Repository, the suzanne mesh (monkey) comes from Blender and the other meshes are of free distribution.

## REFERENCES

- [1] Pierre Alliez, Mark Meyer, and Mathieu Desbrun. Interactive geometry remeshing. *ACM Trans. Graph.*, 21(3):347–354, 2002.
- [2] Pierre Alliez, Giuliana Ucelli, Craig Gotsman, and Marco Attene. *Shape Analysis and Structuring*, chapter Recent Advances in Remeshing of Surfaces, pages 53–82. Mathematics and Visualization. Springer Berlin Heidelberg, 2008.
- [3] Pierre Alliez, Éric Colin de Verdière, Olivier Devillers, and Martin Isenburg. Isotropic surface remeshing. In *SMI '03: Proceedings of the Shape Modeling International 2003*, page 49, Washington, DC, USA, 2003. IEEE Computer Society.
- [4] F. Betul Atalay and David M. Mount. The cost of compatible refinement of simplex decomposition trees. In *IMR*, pages 57–69, 2006.
- [5] Vincent Francois, Jean Christophe Cuilliere, and Michel Gueury. Automatic meshing and remeshing in the simultaneous engineering context. *Research in Engineering Design*, 11:55 – 66, 1999.
- [6] Pascal J. Frey. About surface remeshing, 2000.
- [7] J. Jimenez, F. Feito, R. Segura, and C. Ogayar. Particle oriented collision detection using simplicial coverings and tetra-trees. *Computer Graphics Forum*, 25(1):53 – 68, 2006.
- [8] Juan Jose Jimenez, Francisco R. Feito, and Rafael Jesus Segura. Robust and optimized algorithms for the point-in-polygon inclusion test without pre-processing. *Comput. Graph. Forum*, 28(8):2264–2274, 2009.
- [9] Juan José Jiménez, Rafael Segura, and Francisco Feito. A robust segment/triangle intersection algorithm for interference tests. efficiency study. *Computational Geometry: Theory and Applications*, 43(1):474–492, 2010.
- [10] Dae-Young Kwak and Yong-Taek Im. Hexahedral mesh generation for remeshing in three-dimensional metal forming analyses. *Journal of Materials Processing Technology*, 138(1-3):531 – 537, 2003.
- [11] Tomas Möller and Ben Trumbore. Fast, minimum storage ray/triangle intersection. *SIGGRAPH '05: ACM SIGGRAPH 2005 Courses*, page 7, 2005.
- [12] Carlos Ogayar, Rafael Segura, and Francisco Feito. Point in solid strategies. *Computers & Graphics*, 26(1):616–624, 2005.
- [13] Juan J. Pombo, Jose C. Cabaleiro, and Tomas F. Pena. Parallel complete remeshing for adaptive schemes. *Int. J. Comput. Sci. Eng.*, 1(2-4):207–215, 2005.
- [14] Emil Praun. Spherical parametrization and remeshing. *ACM Trans. Graph.*, 22:340–349, 2003.
- [15] Hanan Samet. *Foundations of Multidimensional and Metric Data Structures (The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.
- [16] Vitaly Surazhsky and Craig Gotsman. Explicit surface remeshing. In *SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 20–30. Eurographics Association, 2003.

# POSTER:Construction of Panoramic Depth Image

Koji Nishio  
Osaka Institute of Technology  
1-79-1, Kitayama  
Hirakata, Osaka  
Japan(573-0196)  
nishio@is.oit.ac.jp

Keiko Hiraoka  
Osaka Institute of Technology  
1-79-1, Kitayama  
Hirakata, Osaka  
Japan(573-0196)  
hiraoka@tipl.is.oit.ac.jp

Chikara Yamashita  
Osaka Institute of Technology  
1-79-1, Kitayama  
Hirakata, Osaka  
Japan(573-0196)  
yamashita@tipl.is.oit.ac.jp

Ken-ichi Kobori  
Osaka Institute of Technology  
1-79-1, Kitayama  
Hirakata, Osaka  
Japan(573-0196)  
kobori@is.oit.ac.jp

## ABSTRACT

We propose a method which constructs a panoramic depth image. We use a digital still camera and a panoramic lens. The lens has omnidirectional mirror that is pointed in the camera direction and the camera can shoot a horizontal panoramic image through it. We use two panoramic images to construct a panoramic depth image. Each image is photographed at different height and there is the parallax difference between these images. Our method seeks corresponding point between these images and calculates the depth value at each pixel. In addition, we had some experience to evaluate effectiveness of our method. The results shows that our method is useful for making three-dimensional contents form the real scenery.

## Keywords

Computer graphic, image processing, panoramic image, depth image, three-dimensional reconstruction.

## 1. INTRODUCTION

In recent years, flat panel displays such as LCD and plasma display have reached mass. Particularly these few years, 3D televisions are launched on the market by various manufacturers and the demand for 3D contents is increasing.

The easy way to get 3D contents is to record the real scenery with a stereo camera. Another way is to make three-dimensional objects and render them using some computer graphic software. However, such software demands the specialized knowledge of computer graphics and the operation skill. In addition, it requires much production time to construct virtual scenery. Then, the techniques constructing the virtual

sceneries from photographed images taken by a camera have gotten a lot of attention recently. With this technique[Jou94a][Tra89a][Tra92a], the virtual scenery can be built up easily because it doesn't require the specialized knowledge about computer graphics. However it requires a huge number of shots to build up detailed virtual sceneries by using a conventional camera. Because of this reason, the method by using the omnidirectional mirror has gotten a lot of attention recently. With this mirror, a number of shots can be reduced because of its wide field of view.

When using computer graphics, the virtual scenery can be viewed stereoscopically by using the depth information preserved at each pixel. However it is necessary to give the depth information to the photographed image in order to view them stereoscopically because the photographed image doesn't have the depth information. The image with the depth information can be generated by using a conventional stereo camera. However there is no established standard method to reconstruct the depth information from photographed images through 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.

omnidirectional lens. In the present report, we propose a method which generates the image with depth by storing the depth information to the omnidirectional image. Our method makes it easy to generate the virtual scenery from photographed omnidirectional images. We had some experiment to evaluate our method. In the experiment, we use a conventional digital still camera SONY DSLR-A900 and an omnidirectional mirror EIZOH Wide70 as shown in figure 1. Figure 2 shows the image photographed by this system.



Figure 1. Omnidirectional mirror

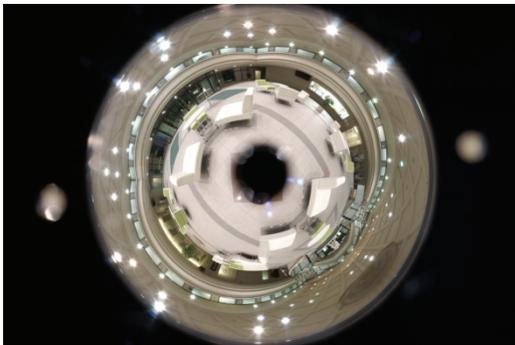


Figure 2. Omnidirectional image

## 2. OFFSET VERTICAL VISION

Our method uses two omnidirectional images and calculates depth information by applying stereo vision to these images. Stereo vision is a method which reconstructs the three-dimensional information from two images photographed at a different view point as shown in figure 3.

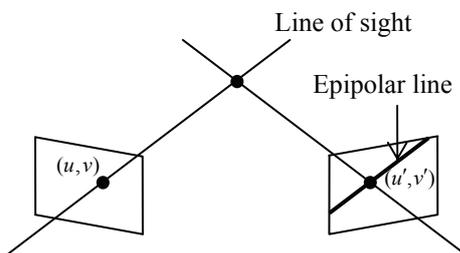


Figure 3. Stereo vision

To reconstruct the three-dimensional information on the principle of stereo vision, it is necessary to find the corresponding point between two images. These corresponding points are found with the constraint that spatial positions correlated to the pixel are on a line. The points corresponding to the pixel on an

image plane are on the line of sight which projected to the other image plane. This line is called epipolar line. Then we fixed the camera to the tripod and take shots elevating the camera. This is called offset vertical vision and we define the image photographed at the upper view point as the upper image and the image photographed at the lower view point as the lower image. We can take the upper image and the lower image at a time with a single camera. In addition, there is vertical translation which is called vertical parallax between those images as shown figure 4.

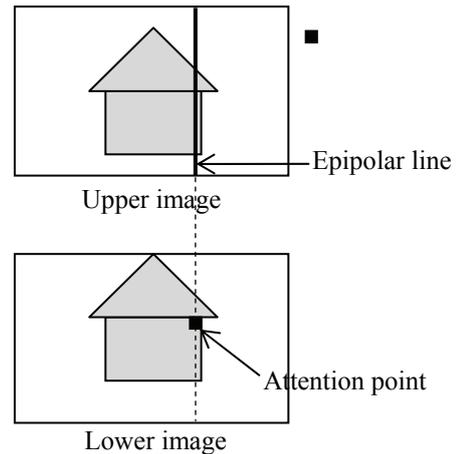


Figure 4. Offset vertical vision and its parallax

## 3. DEPTH CREATION

Our method generates the panoramic depth image from the omnidirectional image which photographed through the omnidirectional mirror. First, the upper image and the lower image are converted into panoramic images by a development projection. Finally, the depth information is calculated for each pixel from the parallax between developed images. However the process may find incorrect corresponding points by mistake. Then we set a limit to the search range for avoidance of mistake.

### Searching for the Corresponding Point

Our method uses template matching to find corresponding points. Template matching searches for the point which is similar to the template image. The process generates the template image from the lower image and finds the corresponding point by searching for the similar region in the upper image.

The corresponding point is found in each image photographed at the lower and the upper view point and the depth information for each pixel is calculated from the parallax of the corresponding point. We applied SSD(sum of squared differences) to calculating the similarity measure. This value is calculated by

$$SSD = \sum_{i=0}^m \sum_{j=0}^n (I(x_i, y_j) - T(x_i, y_j))^2$$

$I$  is the target image and  $T$  is the template image.  $(x_i, y_i)$  is the coordinate of a point in an image.  $m$  and  $n$  are size of target and template image. As SSD approaches to zero, the target image is similar to the template.

### Limitation of the Searching Area

Our method uses the template matching with color information. If the process searches over whole the image, the possibility of miss search is increased at the regions where similar color is. Then we adopt a limitation of the searching area. We use the epipolar lines, the property of the offset vertical vision and the standard deviation of color. First, as described in Chapter 2, the point in an image is on the epipolar line in the other image. Then the searching area can be limited to the epipolar line and the area around it. Second, objects in the lower image come out below the level of corresponding height in the higher image. Then the searching area can be limited to the area below the line that belongs to the top edge of the template. With these limitation, the searching area is reduced as shown in figure 5.

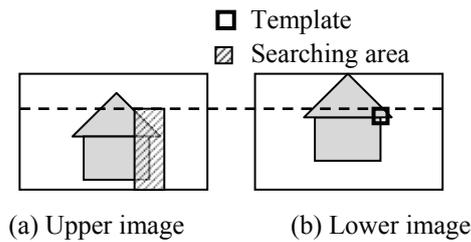


Figure 5. Limitation of searching area

However, the process may find more than one matching result when there're not enough textures in the image. Then if the standard deviation of the template image is not enough high, the process doesn't proceed the template matching for the template because there're not enough textures.

### Calculating the depth value

Our method calculates the depth value by triangulation. The camera position of the upper image is  $A$  and the camera position of the lower image is  $B$ . It is necessary to measure  $l$ ,  $\alpha$ ,  $\beta$ , and  $r$  as shown figure 6.

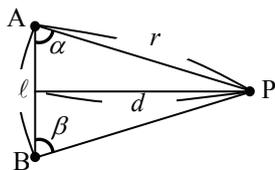


Figure 6. Triangulation

Where the vertical angle of view is  $\theta_h$  and the corresponding points on the images are  $i$  and  $j$ , the height of the image is  $h$ . The  $\alpha$  and  $\beta$  are calculated by

$$\alpha = \begin{cases} \frac{\pi}{2} + \tan^{-1} \left( \left( 1 - \frac{2i}{h} \right) \tan \left( \frac{\theta_h}{2} \right) \right) & \left( 0 \leq i < \frac{h}{2} \right) \\ \frac{\pi}{2} - \tan^{-1} \left( \left( \frac{2i}{h} - 1 \right) \tan \left( \frac{\theta_h}{2} \right) \right) & \left( \frac{h}{2} \leq i < h \right) \end{cases}$$

$$\beta = \begin{cases} \frac{\pi}{2} - \tan^{-1} \left( \left( 1 - \frac{2j}{h} \right) \tan \left( \frac{\theta_h}{2} \right) \right) & \left( 0 \leq j < \frac{h}{2} \right) \\ \frac{\pi}{2} + \tan^{-1} \left( \left( \frac{2j}{h} - 1 \right) \tan \left( \frac{\theta_h}{2} \right) \right) & \left( \frac{h}{2} \leq j < h \right) \end{cases}$$

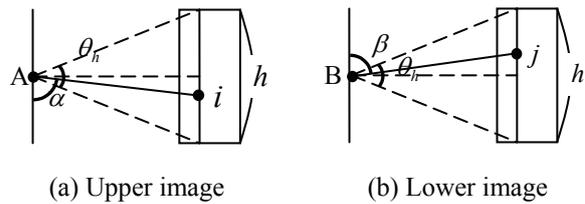


Figure 7. Relationship of the corresponding points and angles

$r$  is calculated by

$$r = \frac{\ell \sin \beta}{\sin(\pi - (\alpha + \beta))}$$

The depth  $d$  as shown in figure 5 is calculated from the trigonometric ratio by

$$d = r \sin \alpha$$

### Depth Value Correction

Some area is not searched as a result of the limitation of the searching area. Incorrect corresponding points may be found even if the searching area is limited. This reads to an incorrect depth value. Then we adopt a depth value correction. Unsearched area doesn't hold the depth value. First, these depth values are interpolated from neighboring pixels up and down as shown in figure 8.

7	8	5
5	-	5
3	-	5
1	2	5

7	8	5
5	6	5
3	4	5
1	2	5

(a) Lack of depth (b) Vertical interpolation

Figure 8. Depth value correction

Second, some depth is differ greatly from one of surrounding pixels. Then we apply median filter to every depth.

#### 4. RESULT AND CONCLUSION

We had some experiments to evaluate the effectiveness of our method. In this experiment, depth image was generated from a pair of omnidirectional images as shown in figure 9(a) and (b). The resolution of each image is  $10,800 \times 3,024$ . Figure 9(c) shows the result generated from these images. Where the bright area means near and the dark area means far. From this result, it is thought that our method can create the depth image almost successfully. However, some area results in black which means far despite near regions. In these areas, incorrect corresponding points are found by mistake. It is thought that this problem can be solved by setting appropriate aperture value to the camera and by using structured lights[Con03a].

#### 5. REFERENCES

[Jou94a] Subbarao, M., Surya, G. Depth from

defocus: A spatial domain approach. *International Journal of Computer Vision* 13, No.3 pp. 271-294, 1994.

[Tra89a] Hoff, William, Ahuja, Narendra. Surfaces from stereo: Integrating feature matching, disparity estimation, and contour detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11, No.2 pp.121-136, 1989.

[Tra92a] Adelson, Edward H., Wang, John Y.A. Single lens stereo with a plenoptic camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14, No.2, pp.99-106, 1992.

[Con03a] Scharstein, D., Szeliski, R. High-accuracy stereo depth maps using structured light. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 1, pp. I/195-I/202, 2003



(a) Upper image



(b) Lower image



(c) Created depth image

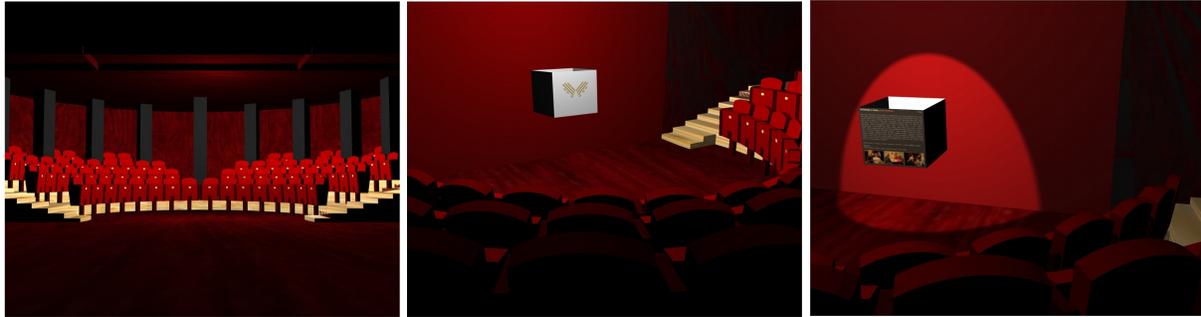
Figure 9

# 3D extension of web

David Běhal  
Comenius University,  
Bratislava, Slovakia  
e-mail: behal@dejw.sk

Jana Dadová  
Comenius University,  
Bratislava, Slovakia  
e-mail: dadova@sccg.sk

Ivana Uhlíková  
Comenius University,  
Bratislava, Slovakia  
e-mail: ivanauhlikova@gmail.com



## ABSTRACT

In our work, we propose our application of 3D graphics inside the web browser. Nowadays, internet and web browsers are not only space, where user can read information, but it is space also for collaboration and experience. Therefore we propose new ideas how should browsing a modern web page look like with use of state-of-the-art technology and possibilities. Users expect from the displayed application more interesting features, not only display of static text. Standard layout of a web page is based on list of a newspaper or a book. This layout is obsolete and does not fit modern approach in new technologies. Our method of displaying content and information inside the web browser is with full 3D, not only embedding 3D as a special feature inside a small part of web page. In this article we also include virtual Theatre as a case study for this problem and explain possibilities of information display and reasons why is 3D better than 2D in this case.

**Keywords:** 3D web, VRML, X3D, browser

## 1 INTRODUCTION

These days there are more possibilities how to make interactive and interesting application for web pages. The web page, as we know is today 2D content derived from the news page. Although there are some possibilities to put 3D content inside a web page with the help of plug-in, most of the web pages are still 2D. In our work we are presenting some ideas how to extend 2D web page to a 3D space in novel way. We also provide a case study where we show that it can be really useful to use 3D space instead of standard 2D web page.

Our work is divided into 3 parts. In the first part we will focus on 3D interface for a web page and our way of transferring 2D interface from standard web page into 3D space. The second part describes the case study. This study shows that in case of cinema or theatre it is possible to use 3D space and also the user or visitor will have better experience and fun. Last section brings plans to the future and conclusion.

## 2 BACKGROUND AND RELATED WORK

In this section, we will discuss different ways of displaying a 3D content inside the web browser. First we have to say, that none of today web browsers have implemented direct support for 3D content, only with the use of plugin. The HTML version 4.01 and also XHTML specification do not have the support for 3D content [W3C99], [W3C00]. The same is for styling the document. Cascading style sheet (CSS) version 2

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.

have support just for styling 2D content [W3C09].

On the other hand, there are some formats, that are used to present 3D content inside web page. The oldest one is Virtual reality modeling language (VRML). Advantages are: it is an open standard, ready to be shared on the internet and have direct support for interaction [vBSF04]. Disadvantages are: it is old format and lacks some new features, plugin for the browser is needed.

Newer and popular formats, similar to VRML are COLLADA and X3D. Both are based on XML and divided in to a profiles for better usability and both do not support direct way of interaction [wDc05], [AB06]. First one is newer version of VRML and become a standard for 3D virtual space. VRML and X3D are the most used format for 3D on web [wDc]. Nevertheless direct interaction is not supported in X3D, it is possible to use combination of X3D and AJAX as AJAX3D [Par07]. Second one is primary used for data exchange and nowadays browsers do not have support for displaying it, there are no plugins. One option is Google's project O3D [Goo10], that adds interaction with use of JavaScript API and display COLLADA with use of plugin from Google.

Above mentioned formats need a plugin for a display. On the other hand, future will probably bring new options. WebGL is a new web standard for low-level 3D graphics. It is based on OpenGL ES 2.0 and uses the canvas element in HTML5 [W3C10], that means that you do not need to use any plug-in. However the HTML5 is not a standard yet, so the support for WebGL will be in the next generation of web browsers [Khr10].

Beside the mentioned technologies, there are a lot of other projects, which try to extend web page into 3D. One of them is X3DOM, where the authors propose to use X3D and integrated it in DOM tree. Then it is easy to manipulate with 3D objects [BEJZ09]. Another project is 3DMLW, which is a technology for creating and displaying 3D and 2D content on the web through common web-browsers [tR08]. Authors of [Pau10] proposed to use 3DCSS to separate visual properties from other properties in 3D scene. The 3DCSS is based on CSS 2.1. They also propose to use 3D MARKUP SCENE DESCRIPTION (3DMSD), which extends the current web page model in 3D.

Using 3D in the web page is not novel. Few of the projects in cultural heritage use also 3D content to present their exhibits at whole museum [Fer07] or [moL05]. Nevertheless, these projects and solution use 3D only as extension in part of the page, to show some scenes, or some objects. We would like to focus on 3D web pages, that do not display anything from the WSCG 2011 Poster Papers

content of the page outside of the 3D plugin.

### 3 3D INTERFACE

In this section we propose our solution how to create a 3D web page and how to create a simple web page interface in 3D. For our solution we use VRML or X3D, but in future we would like to convert it into WebGL.

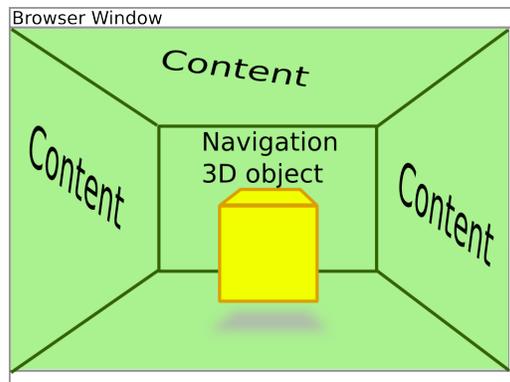


Figure 1: Our proposed solution of 3D web browser instead of 2D with navigation cube.

In 2D space the author of a web page has just one plane to use. Most of the web pages has the similar layout. If we extend the web page into 3D space we get much more space where to put information, content etc, see fig. 1. Screenspace is not changed, just the way how to display content. As the 3D web pages are not so spread, we would like to propose how the interface can look like. Instead of pages we can use room as a part of whole site. Unlike SphereXP by Hamar [Ham05] and Task Gallery [Rob09], our 3D GUI approach does not open multiple application windows.

Navigation in this kind of web page is also very important. Users are not so familiar with 3D on web, so the navigation should be clear enough and it should be based on real world or on common navigation in 2D web pages. We decided for the object similar to one in real world. For the site with more than one room, we can take analogy from real world and use doors as the links between rooms. However, the user has to have the information where the doors will take him. Our scene is different, we use only one room and the scene is dynamically changed. We use a cube, which contains all setting and whole menu for the site. The cube is the central point to navigate (see fig. 1). This cube can show also the text information or text content of the site. .

## 4 CASE-STUDY - THEATRE

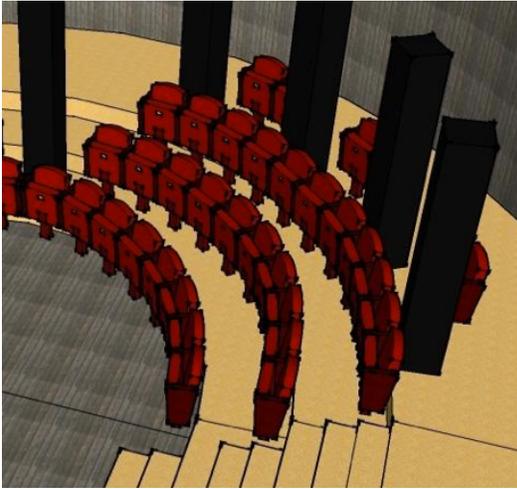


Figure 2: *The part of the John Palarik Theatre in Trnava, Slovakia. (Screenshot from our application.)*

In this section we will show you our solution in real data. In section 3 we discussed use of 3D vs. 2D. One of the good examples, where 3D page has more potential than 2D is a theatre, because for the user it could be interesting to have a feeling that he is inside, instead of reading a brochure about it (analogy to 2D page). Our theatre is created based on real John Palarik Theatre in Trnava, Slovakia (fig. 2.)

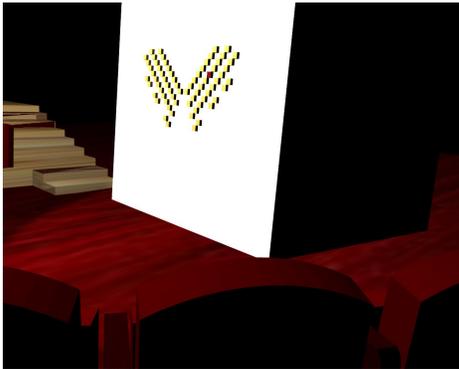


Figure 3: *User is looking at the scene from the selected seat.*

It is the room with stage and spectators (see teaser). On the stage you can find our cube mentioned before (section 3). In this case the cube represent the navigation menu and also it shows some information (see teaser). The user can interact with this cube. We choose rotation around up vector, that means we will use only 4 faces from cube. Three faces contain information about theatre as in 2D page, but fourth face is used specifically for 3D. On this face the user can find the WSCG 2011 Poster Papers

ground plan of the theatre with all seats available. Visitor can choose any seat and the view will be transformed into the position of that seat. This gives to the user the exact look into a scene as in the real theatre.



Figure 4: *Price displayed on the seats.*

The user is also able to click on each seat and avatar [vBSF04] will be also transported into that place (fig. 3). Afterwards user can find out the angle of the view and also if there are some barriers like columns etc. Even price for the specific seat could be displayed directly on the seat (fig. 4). When the view is satisfactory it is also possible to proceed to checkout and buy the tickets. We propose this solution in 3D, because user will have better idea of the building structure and possible view from the seat, which in plain 2D will not be satisfactory.

It is possible to use theatre in 3D not only for buying tickets, but for the show itself. We could add virtual performance, or video screen on the virtual stage and bring user real feeling from the theater. This could help user even more to choose a seat. Moreover also additional features are possible during show, which are not common in real theaters, but could be useful such as subtitles.

### 4.1 Other possible case studies

The virtual theatre is not the only one possibility when is better to use 3D instead of standard 2D web page. It could be used also in cinemas or e-shops. If the user came to the e-shop, products are displayed same way as in catalogue. But in case of 3D site, user get the opportunity to see the 3D model of that object and can also interact with it, can play with it. Shopping could be even more fun, similar to playing a game. This way could help shop to have more customers that like to spend some time in a shop and shopping is entertainment for them. A lot of people are visiting shops to get some inspiration, so the user can do that also in 3D e-shop.

The approach mentioned above is better theoretical and real user feeling need to be better evaluated. We created prototype, but to prove our assumptions we need to create deeper user study. This we leave to the future work by now.

## 5 CONCLUSION AND FUTURE WORK

In our work we show possibility of layout change for some modern web pages. We proposed solution, where 3D space is displayed in whole internet browser space instead of just part. Our case-study shows how this approach can be used for the better user orientation and comfort. Moreover, in this space even more information can be added. In future work we would like to convert our solution and case study into WebGL to get a direct support from the web browser, not only use of a plugin. Another important improvement would be better user study as is mentioned above sec. 4.1.

## 6 ACKNOWLEDGEMENT

This research is supported by Slovak Ministry of Education, VEGA grant No. 1/0763/09.

## References

- [AB06] Remi Arnaud and Mark C. Barnes. *Colлада: Sailing the Gulf of 3d Digital Content Creation*. AK Peters Ltd, 2006.
- [BEJZ09] Johannes Behr, Peter Eschler, Yvonne Jung, and Michael Zöllner. X3dom: a dom-based html5/x3d integration model. In *Web3D '09: Proceedings of the 14th International Conference on 3D Web Technology*, pages 127–135, New York, NY, USA, 2009. ACM.
- [Fer07] Andrej Ferko. Virtual museum technologies. virtual university. [ONLINE] (10.9.2009) <http://www.sccg.sk/~ferko>, 2007.
- [Goo10] Google. WebGL implementation of o3d. [ONLINE] (27.9.2010) <http://code.google.com/p/o3d/>, 2010.
- [Ham05] Dusan Hamar. Spherexp portal. [ONLINE] (3.1.2011) <http://www.spheresite.com/>, 2005.
- [Khr10] Khronos. WebGL - opengl es 2.0 for the web. [ONLINE] (27.9.2010) <http://www.khronos.org/webgl/>, october 2010.
- [moL05] The museum of Louvre. The museum of louvre. [ONLINE] (20. 9. 2009) <http://www.louvre.fr>, 2005.
- [Par07] Tony Parisi. Ajax3d - the open platform for rich 3d web applications. [ONLINE] (27.9.2010) <http://opensource.sys-con.com/node/336895>, May 2007.
- [Pau10] Peter Paulis. 3d webpages. Master's thesis, Comenius University, Bratislava, Slovakia, May 2010.
- [Rob09] Daniel C. Robbins. Taskgallery home page. [ONLINE] (3.1.2011) <http://research.microsoft.com/en-us/um/people/dcr/work/TaskGallery/>, 2009.
- [tR08] 3D technologies R&D. 3d markup language for web. [ONLINE] (10.10.2010) <http://www.3dmlw.com/>, 2008.
- [vBSF04] Jiří Žára, Bedřich Beneš, Jiří Sochor, and Petr Felkel. *Moderní počítačová grafika*. Computer Press, 2004. ISBN 80-251-0454-0.
- [W3C99] W3C. Html 4.01 specification. [ONLINE] (27.9.2010) <http://www.w3.org/TR/REC-html40/>, December 1999.
- [W3C00] W3C. Xhtml 1.0 the extensible hypertext markup language (second edition). [ONLINE] (27.9.2010) <http://www.w3.org/TR/xhtml1/>, January 2000.
- [W3C09] W3C. Cascading style sheets level 2 revision 1 (css 2.1) specification. [ONLINE] (27.9.2010) <http://www.w3.org/TR/CSS2/>, September 2009.
- [W3C10] W3C. Html5. [ONLINE] (27.9.2010) <http://www.w3.org/TR/html5/>, october 2010.
- [wDc] web 3D consortium. X3d, vrml, the most widely used 3d formats. [ONLINE] (3.1.2011) <http://www.web3d.org/realtime-3d/x3d-vrml/x3d-vrml-most-widely-used-3d-formats>.
- [wDc05] web 3D consortium. X3d international standards. [ONLINE] (27.9.2010) <http://www.web3d.org/x3d/specifications/>, 2005.

# Interactive Bivariate Mode Tree

Martin Florek

Comenius University, Department of Applied Informatics  
Bratislava, Slovakia  
florek@sccg.sk

Helwig Hauser

University of Bergen, Department of Informatics  
Bergen, Norway  
Helwig.Hauser@uib.no

## ABSTRACT

Kernel density estimation (KDE) is widely used statistical method to study distribution of the data. The problem with this method is choosing the right bandwidth. In our work we focus on semiautomatic bandwidth selection with the use of visual paradigm of the Mode Tree. We not only enhanced the bivariate mode tree visualization, which was only sketched by the authors, but we also developed a hybrid CPU/GPU implementation to improve the speed of the mode tree construction.

**Keywords:** Mode Tree, Data Visualization, Kernel Density Estimation, GPU.

## 1 INTRODUCTION

Nonparametric density estimation was overlooked for years as an interactive visual data exploration tool. It was mainly due to the intense computational complexity, but with today's GPU algorithms even a low end graphics hardware can compute a kernel density estimate in real time. Current hardware allow us to interactively change the parameters and see the results instantaneously, but the search for the best bandwidth is not easily solved even if we can compute and observe tens of different KDEs in real time.

Thus in our work we picked up the mode tree visual paradigm and brought it to interactive life with a GPU algorithm. This powerful tool is not widely used, because one needs to compute tens to hundreds of different KDEs, which is a time consuming task. The Mode Tree is a powerful graphical tool for choosing the right bandwidth for KDE construction, it can reveal distributional characteristics, such as multimodality. Here we present an implementation which is capable of constructing a bivariate mode tree on a main stream hardware in interactive rates and we also enriched the visualization with following enhancements:

- height and depth augmented visualization for easy identification of important features and suppressing the outliers
- mitigated clutter of the 3D plot
- interactive hybrid CPU/GPU implementation

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.

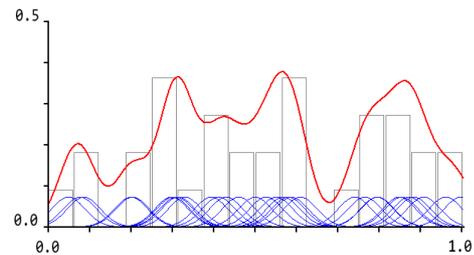


Figure 1: 1D kernel density estimate (in red) for 32 random samples with the bandwidth of 0.04 with the corresponding Gaussian kernels (in blue) and a 16-bin histogram (in grey).

## 2 RELATED WORK

KDE (Fig. 1) is a well established visual analysis method in many fields, e.g. in bioinformatics [2, 3] and in visualization of heat maps [5]. This idea was also adapted to the parallel coordinates [4] visualization [21, 1].

In nonparametric density estimation, the kernel function choice does not influence the final density reconstruction that much. More important is the choice of the right bandwidth, the smoothing parameter. This choice effects how much of the underlying structure will be exposed to the user. Wrong bandwidth can lead from undersmoothed to oversmoothed reconstruction, as can be seen in Fig. 2.

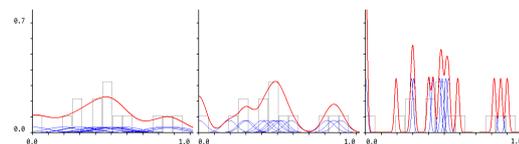


Figure 2: Three KDEs from random data with varying bandwidths from oversmoothed to undersmoothed results.

Scientists were trying to find automatic methods to find the best bandwidth [18], some say that the bandwidth should vary with data sample [10]. In the adaptive bandwidth selection, the work by Maciejew-

ski [7] uses the  $k$ -nearest neighbours to estimate the kernel. Until now, there is no completely satisfactory automatic method to find the right bandwidth and many real world data show, that one global bandwidth is insufficient.

For this reason a visual tool called the Mode Tree was introduced by Minnotte and Scott [11]. This method summarizes information from density estimates computed for many different bandwidths  $h$  (Sec 4). The mode tree was further enhanced to display modes' sizes, antimodes and bumps. Later Marchette and Wegman introduced the Filtered Mode Tree which is constructed from filtered kernel estimator [8]. Minnotte et al. also developed a Mode Forest [9], which is a simultaneous view of many mode trees, which are based on original data but with variations such as resampling and jitter. This method can filter out outliers which can strongly affect the simple mode tree.

Recent research on mode trees was done by Klemalä [6] who generalized the idea of mode tree to multivariate settings in form of multiframe mode graphs.

### 3 KDE

Kernel density estimation (KDE) was introduced by Rosenblatt [14] and Parzen [12]. Given a set of  $n$  data samples  $(x_1, \dots, x_n) | x_i \in \mathbf{R}$  the KDE is

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x-x_i}{h}\right) \quad (1)$$

where  $K$  is a kernel and  $h$  is the bandwidth (smoothing parameter). The kernel function  $K$  usually has the following properties:  $K(x) \geq 0 \forall x$ ,  $\int_{-\infty}^{\infty} K(x)dx = 1$  and  $K$  is centered in 0. In our work we focus on the Normal kernel and figure 1 shows an example of a KDE with the corresponding Gaussian kernels and histogram.

#### 3.1 2D KDE

Focus of this work is on the bivariate mode tree, so we need an extended 2D kernel to compute 2D density estimate. Equation 1 for two dimensions is

$$\hat{f}_{\mathbf{H}}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n K_{\mathbf{H}}(\mathbf{x} - \mathbf{x}_i) \quad (2)$$

where  $\mathbf{x} \in \{(x,y) | x,y \in \mathbf{R}\}$  and  $\mathbf{H}$  is a symmetric positive-definite bandwidth matrix. In our work we have decided to use two different bandwidths (each for one dimension) and not to use the full potential of the matrix  $\mathbf{H}$ , which would also allow rotation of the kernel. The final separable extension of equation 1 to 2D with two bandwidths looks like this (Eq. 3) and an example 2D plot can be seen in figure 3.

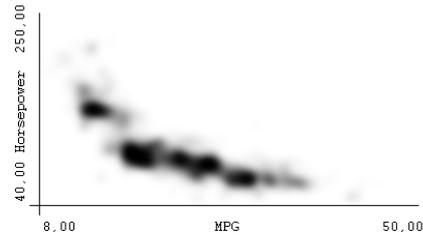


Figure 3: Example of a two dimensional KDE from the cars dataset [20].

$$\hat{f}_h(x,y) = \frac{1}{n} \sum_{i=1}^n K\left(\frac{x-x_i}{h_x}\right) \cdot K\left(\frac{y-y_i}{h_y}\right) \quad (3)$$

For details on how to derive the final equation 3 we refer to Scott and Sain [15], where extensions to arbitrary dimensions can be found.

## 4 MODE TREE

For 1D KDE the mode locations are plotted against the bandwidth at which the density estimate with those modes is calculated [11]. In Fig. 4 the solid vertical lines represent the modes corresponding to those in the density estimates. Important thing in the mode tree construction is to use a logarithmic scale for choosing the different values of  $h$ , because at high values of  $h$  large changes have lesser effect on the KDE than smaller changes at lower values of  $h$ . Plotted locations of all modes result in a set of lines called mode traces.

Silverman [16] proved, that for the normal kernel the number of zeros in all derivatives of  $\hat{f}_h$  is decreasing for increasing  $h$ . This implies that also the number of modes is decreasing as the bandwidth is increasing.

## 5 2D MODE TREE

The mode tree is very useful tool in one dimensional case and has a potential to be good with 2D KDE. In the bivariate case it becomes a 3D plot instead of 2D, where on  $X$ -axis is the first dimension, on  $Y$ -axis is the second dimension and the  $Z$ -axis represents increasing bandwidths and the streaks represent modes' locations (Fig. 5).

Minnotte et al. in their work only sketched the idea of bivariate mode trees [11]. They also had to resample the data to have variance equal to 1, because they used only single bandwidth for both dimensions. In our work we do not have a limitation of resampling the data and of single bandwidth.

One of the biggest drawbacks of mode trees is their computational complexity, which is even bigger problem in the bivariate case. Recent main stream GPUs

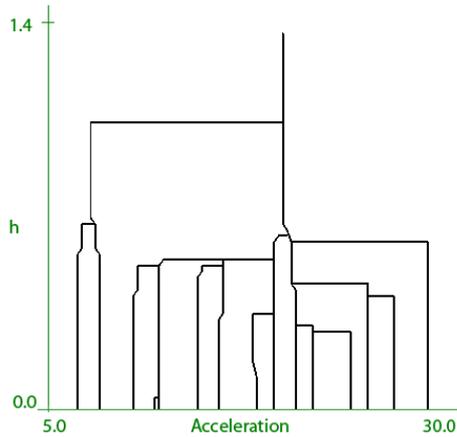


Figure 4: 1D Mode Tree. Solid vertical lines represent the modes corresponding to those in the density estimates, where the bandwidth is on the Y-axis. Horizontal lines are links from extinct modes to the modes, which consumed them. Plotted from 100 KDEs.

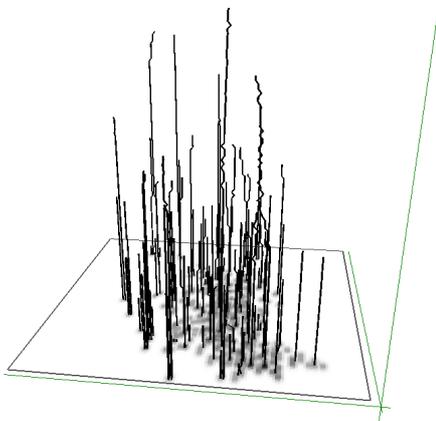


Figure 5: The bivariate Mode Tree as defined by Minnotte and Scott [11]. The streaks represent modes' locations on 2D KDE for increasing bandwidths (the Z-coordinate). At the bottom of the mode tree there is the 2D KDE of the lower bandwidth limit.

can compute the 2D KDE in interactive times and we modified the current GPU algorithm to be even more efficient for the mode tree computation (Sec. 7.1).

## 6 ENHANCED BIVARIATE MODE TREE

In this work we focus on enhancing the bivariate mode tree, not only to mitigate the problems of three dimensional plots (occlusion, navigation, etc.) but also make the visualization more informative and clear to the user. We also developed a GPU implementation to compute the 2D mode tree and for smaller data sets we achieved instantaneous construction times (Sec. 7.1).

One very important information, which is not communicated to the user in the original mode tree, is the

height of the modes. The mode tree tells us only where the modes are and nothing more. To address this issue we used special color coding with monotonically increasing luminance of the color (Fig. 6), as developed by Wyszecski and Stiles [22].



Figure 6: Color coding with increasing luminance used to encode mode's height.

The color and the brightness represent the relative height of the modes, where the black color represents the tallest mode and streaks with lighter colors are the smaller modes, possible outliers. This color coding makes the significant modes stand out, while the small modes, outliers, are more hidden from the viewer.

Another problem of all 3D plots, not only bivariate mode tree plot, is occlusion, so the features closer to the camera occlude features farther away. This can lead to hiding important features behind unimportant ones, in some camera positions. To mitigate the problems with occlusion, we used semi transparent mode streaks, where the transparency values are based on the relative height of the mode to the current tallest mode. This means that small modes are represented not only by lighter streaks (which could occlude dark streaks), but the streaks are also semi transparent (Fig. 7). The smaller the mode, the more transparent it is, thus ensuring that significant modes represented by dark streaks are always clearly visible and only subtly overlaid by the outliers. If the user wants to study outlying modes, then the color coding and semi transparency can be inverted.

The last problem of the 3D plots we addressed in this work is difficult spatial location deduction. When looking at a simple bivariate mode tree (Fig. 5) it is hard to deduce the correct locations. Even an approximate location is very hard to guess. We adapted a simple idea from [13], where they made contour thickness of a point in space dependent on distance to the camera. So we made the mode streaks closer to the camera more thicker and the far away streaks are thinner, which improves depth perception (Fig. 7).

## 7 IMPLEMENTATION

Our implementation is a combination of a CPU and GPU algorithm. The KDEs are computed on the GPU (Sec. 7.1) and the identification of modes and mode traces construction is done on the CPU. Finding modes on a 2D grid is a simple task, but constructing the mode streaks is a more challenging one.

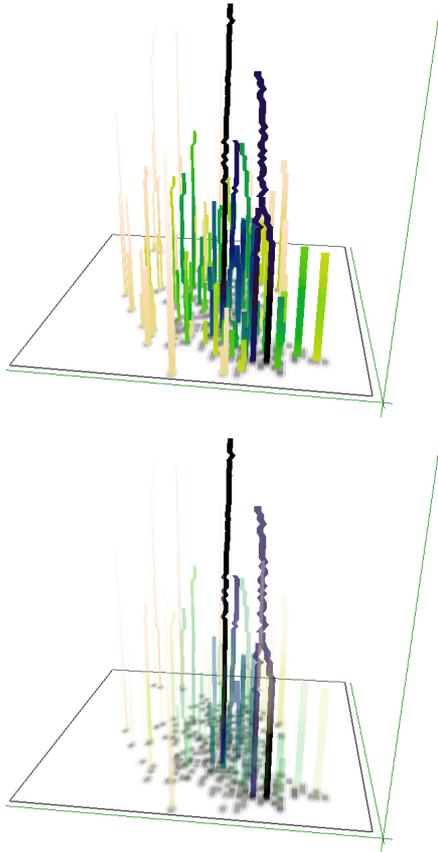


Figure 7: Enhanced bivariate mode tree: (top) Color coding enabled, which reveals relative height of the modes to the tallest mode, but with some small modes occluding taller ones. (bottom) With transparency enabled, mitigating the clutter caused by plotting the mode streaks in 3D. Only the important peaks are clearly displayed and also their starting positions are always visible. Streaks closer to the camera are more thick and the far away streaks are thinner, for better depth perception.

First we start by computing many KDE's with different bandwidths, from zero to the variance of the data. The default value for the number of density estimates was set to 64 and the bandwidth values are equally spaced on a logarithmic scale. For the reason we refer to the section 4. After this we identify the modes and then run a matching algorithm to connect the modes' locations for different bandwidths to construct a mode streak. For this we developed a simple multipass heuristic, where in the first pass we connect modes which are directly above each other and remove these modes from the queue of unprocessed modes. Then we run multiple passes where in each pass we try to find a corresponding mode streak for each unprocessed mode, within an increasing radius.

## 7.1 GPU implementation

Computing hundreds of KDEs is computationally the most intensive task and therefore is entirely done on a GPU. For this we developed a GPU algorithm, where we precompute four Gaussian kernels into an RGBA floating point texture, each for a different bandwidth and to a different texture channel. Then we use a splatting technique, which results in four KDEs computed with different bandwidths, in the same time as one. The thing that slows down the construction the most, is not the KDEs computation, but the construction of mode streaks, which is done on a CPU (Sec. 7).

Table 1 shows performance values for different data sets and different GPUs, from low-end laptop Intel 4 mobile GPU to a powerful AMD HD5700. The times vary for the same GPU, because the construction time is heavily dependent on the point matching algorithm, which depends on the number of modes. We can compute mode tree in real time even on low end hardware.

data / GPU	Intel 4 mobile	AMD HD5700
cars (400)	200 - 300 ms	60 - 90 ms
out5d (16k)	2800 - 3900 ms	600 - 800 ms

Table 1: Time of 2D mode tree construction, with  $128 \times 128$  KDE resolution and 100 different bandwidth values. The cars data set has 400 records and out5d has 16000 records.

## 8 CONCLUSION AND FUTURE WORK

This work brought attention to an old idea of the Bivariate Mode Tree. We achieved not only an interactive hybrid GPU/CPU implementation, but we also enhanced the mode tree visualization with information about the height of the modes for easy identification of important modes. With making the unimportant modes transparent, we removed some clutter, which is one of the biggest problems of 3D plots. We also made modifications for better depth perception of mode streaks for easier spatial localization.

In the future we want to further enhance the plot, for example with better outliers visualization (stippled lines) and with links from extinct modes to the modes which consumed them. Further we would like to improve the performance of our GPU algorithm with the use of multiple render targets, which will allow us to render 16–32 KDEs at once with only a small time increase compared to the current implementation. And the last thing we will do is a better initial camera placement, which is very important in 3D plots, but is not a trivial task. This could be achieved by geometric best view algorithms [19], which we will have to adapt to the lines geometry.

## 9 ACKNOWLEDGEMENT

This work has in part been funded by Slovak Ministry of Education VEGA No. 1/0763/09. Parts of this work have been done in the context of the VisMaster project, which acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission under FET-Open grant number 225429. We would like to thank authors of the Xmdv-Tool [20], to the StatLib [17] web page and especially to prof. Matthew Ward for providing the data sets.

## REFERENCES

- [1] Almir Olivette Artero, Maria Cristina Ferreira de Oliveira, and Haim Levkowitz. Uncovering clusters in crowded parallel coordinates visualizations. In *INFOVIS '04: Proceedings of the IEEE Symposium on Information Visualization*, pages 81–88, Washington, DC, USA, 2004. IEEE Computer Society.
- [2] Paul H. C. Eilers and Jelle J. Goeman. Enhancing scatterplots with smoothed densities. *Bioinformatics*, 20(5):623–628, 2004.
- [3] Florian Hahne, Dorit Arlt, Mamatha Saueremann, Meher Majety, Annemarie Poustka, Stefan Wiemann, and Wolfgang Huber. Statistical methods and software for the analysis of high throughput reverse genetic assays using flow cytometry readouts. *Genome Biology*, 7:R77+, August 2006.
- [4] Alfred Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(4):69–91, Dec 1985.
- [5] Paul Kidwell, Guy Lebanon, and William S. Cleveland. Visualizing incomplete and partially ranked data. *IEEE Trans. Vis. Comput. Graph.*, 14(6):1356–1363, 2008.
- [6] Jussi Klemälä. Mode trees for multivariate data. *Journal of Computational and Graphical Statistics*, 17(4):860–869, December 2008.
- [7] Ross Maciejewski, Insoo Woo, Wei Chen, and David Ebert. Structuring feature space: A non-parametric method for volumetric transfer function generation. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1473–1480, 2009.
- [8] David J. Marchette and Edward J. Wegman. The filtered mode tree. *Journal of Computational and Graphical Statistics*, 6(2):143–159, June 1997.
- [9] Michael C. Minnote, David J. Marchette, and Edward J. Wegman. The bumpy road to the mode forest. *Journal of Computational and Graphical Statistics*, 7(2):239–251, June 1998.
- [10] Michael C. Minnotte and David W. Scott. The mode tree: A tool for visualization of nonparametric density features. *Journal of Computational and Graphical Statistics*, 2:51–68, 1993.
- [11] Michael C. Minnotte and David W. Scott. The mode tree: A tool for visualization of nonparametric density features. *Journal of Computational and Graphical Statistics*, 2:51–68, 1993.
- [12] Emanuel Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.
- [13] Harald Piringer, Robert Kosara, and Helwig Hauser. Interactive focus+context visualization with linked 2d/3d scatterplots. In *In proceedings of the International Conference on Coordinated & Multiple Views in Exploratory Visualization (CMV 2004)*, pages 49–60. IEEE Computer Society, 2004.
- [14] Murray Rosenblatt. Remarks on some non-parametric estimates of a density function. *The Annals of Mathematical Statistics*, 27:832–837, 1956.
- [15] D. W. Scott and S. R. Sain. "Multi-Dimensional Density Estimation", pages 229–263. Elsevier, 2004.
- [16] B. W. Silverman. Using kernel density estimates to investigate multimodality. *Journal of the Royal Statistical Society. Series B (Methodological)*, 43(1):97–99, 1981.
- [17] Statlib. Statlib. <http://lib.stat.cmu.edu/>.
- [18] Berwin A. Turlach. Bandwidth selection in kernel density estimation: A review. In *CORE and Institut de Statistique*, pages 23–493, 1993.
- [19] Pere-Pau Vázquez and Mateu Sbert. Fast adaptive selection of best views. In *ICCSA'03: Proceedings of the 2003 international conference on Computational science and its applications*, pages 295–305, Berlin, Heidelberg, 2003. Springer-Verlag.
- [20] Matthew O Ward, Elke A Rundensteiner, Qingguang Cui, Zaixian Xie, Di Yang, Charudatta Wad, and Do Quyen Nguyen. XmdvTool, 2009.
- [21] Edward J. Wegman. Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association*, 85(411):664–675, 1990.
- [22] Günther Wyszecki and W. S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulae*. Wiley-Interscience, 2 edition, September 1982.



# Improving the Visualization of Galactic Events Using Pixar's RenderMan

Paul Cassidy<sup>1</sup>  
ptc6340@rit.edu

Tyler Kilburn<sup>2</sup>  
kilburn\_tyler@yahoo.com

Vincent Salemink<sup>1</sup>  
vincentsalemink@msn.com

Reynold Bailey<sup>1</sup>  
rjb@cs.rit.edu

Hans-Peter Bischof<sup>1</sup>  
hpb@cs.rit.edu

<sup>1</sup>Department of Computer Science, Rochester Institute of Technology

<sup>2</sup>Department of Computer Science, Saint John Fisher College

## ABSTRACT

Pixar's PhotoRealistic RenderMan® is a powerful graphics rendering system with underexplored potential for visualizing scientific data. We utilized PhotoRealistic RenderMan® to visualize simulations of galaxy mergers which are dynamic and contain large volumes of multidimensional data. Our goal was to emulate existing astronomical imagery such as images captured by the Hubble Space Telescope. In order to accomplish this we developed techniques for rendering individual stars and also took into account factors such as galaxy density. PhotoRealistic RenderMan® was used to create a movie from the provided galaxy merger data. In this paper we describe our approach and present the results. We also discuss how the rendering framework can be extended to incorporate the effects of light bending due to gravity.

## Keywords

Visualization, RenderMan®, Simulation, Galactic events.

## 1. INTRODUCTION

Humans are extremely limited in their ability to process and understand raw scientific data. This is especially true for large datasets. Simulations of galactic events such as black hole mergers for example, generate huge volumes of complex multidimensional data. Scientific visualization is needed to help bring meaning to this information. Computer Generated Imagery (CGI) can be used to display such phenomena. Typically existing tools are used or new ones must be developed. Powerful tools such as Pixar's PhotoRealistic RenderMan® [Kau00] have strong roots in computer animation and in the film industry. However, PhotoRealistic RenderMan® is often viewed by the scientific community as a means of creating fantasy worlds and special effects and has largely been ignored as a resource for visualizing scientific data.

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.

PhotoRealistic RenderMan® has several important characteristics that make it a viable and effective tool for scientific visualization. These include numerous advanced features for creating visual effects and the ability to use distributed processing to take advantage of large multi-core networks of computers to increase overall throughput. In recent years, several planetariums have begun using rendering packages such as PhotoRealistic RenderMan® to create immersive experiences for their visitors [Mol10]. In this paper we demonstrate how PhotoRealistic RenderMan® can be used to visualize simulations of galaxy mergers. Our aim was to create compelling imagery similar to the stunning images captured by the Hubble Space Telescope (see Figure 1).



**Figure 1. Galaxy NGC 4414. Courtesy of NASA and the European Space Agency. Image generated by the Hubble Space Telescope.**

The remainder of this paper is structured as follows: section 2 discusses background and related work; section 3 presents our visualization framework and rendering techniques; results are presented in section 4; we conclude and describe future work in section 5.

## 2. BACKGROUND AND RELATED WORK

The Center for Computational Relativity and Gravitation (CCRG) is a research group that utilizes mathematical modeling, supercomputing and data visualization to help further our understanding of astrophysical phenomena [RIT10]. Researchers at CCRG have developed a visualization framework called Spiegel, which is equipped with a convenient graphical interface for creating programs to extract, analyze, and render different views of the data [Bis05]. Spiegel was originally designed to utilize Java3D or JOGL to create the final images. Recently, Espinal et al. extended the Spiegel framework to utilize PhotoRealistic RenderMan® [Esp10]. They demonstrated the capabilities of the improved framework by creating renderings of galaxy mergers. However they utilized relatively simple shaders, which were not very realistic. Our aim was to create more realistic images that were closer in appearance to imagery captured by the Hubble Space Telescope. To accomplish this goal, we consider the appearance of individual stars and overall star density.

## 3. VISUALIZATION FRAMEWORK AND RENDERING TECHNIQUES

Individual frames of the final video sequence were produced using simulation files provided by the Center for Computational Relativity and Gravitation. The files contain positional data of every star and black hole for each timestep in a galaxy merger simulation. The Spiegel graphical interface was used to establish viewing parameters and a RenderMan® Interface Bytestream (RIB) file was then generated for each frame. RIB files are simply text files that contain a sequence of commands that RenderMan® compliant renderers such as PhotoRealistic RenderMan® can interpret and execute.

In reality, black holes are not visually perceptible because the gravitational pull of black hole is so strong that not even light can escape from its center. Since our focus was on realism, we opted to ignore the presence of black holes in our renderings and instead focused on creating more realistic stars.

Figure 2 shows an overview of our visualization framework. Prior to this work, the stars were typically rendered as simple spheres. For our approach, each star is actually a combination of three spheres with separate shaders. Using a modified version of RenderMan’s built in Perlin-based noise function [Per85] and the built in spline function, the

main surface color across the star was created. For the spline, five colors are used: a base color, usually white or black, and four colors similar to the desired star color. The color palette is shown in Figure 3. To simulate sunspots, we modified the built in noise function and used the RenderMan® mix function to add this into the overall surface color.

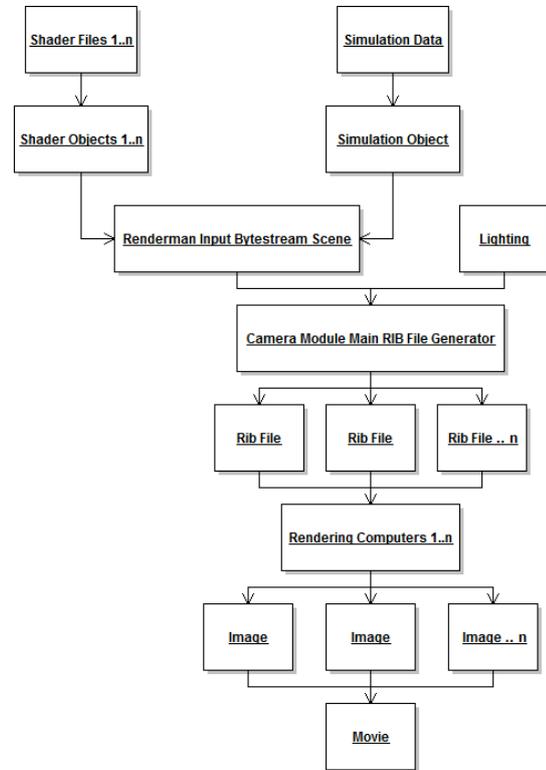


Figure 2: Overview of visualization framework.

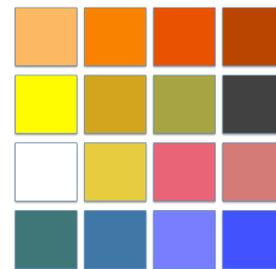
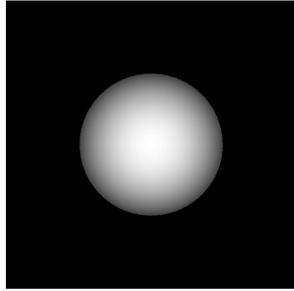
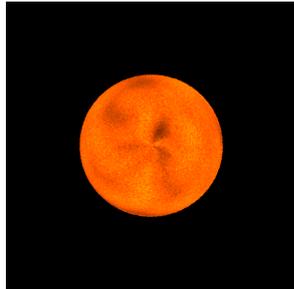


Figure 3: Palette of possible colors for stars.

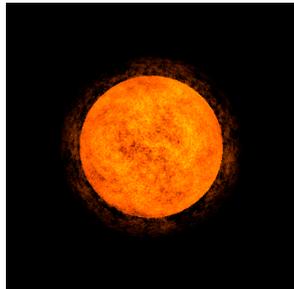
To simulate the prominences of the stars, i.e. a cloud of gas elevated above the star surface, a second sphere is applied over the main sphere and an opaque, fog-like surface shader of the appropriate color is applied. The specific color of the prominence surface shader does not have to be the same as the surface color of a star. For example, a red star can have a yellow prominence and it would still look acceptable. Additionally, a displacement shader is applied to the prominence to further enhance the effect. Figure 4 shows the effect of each of these steps.



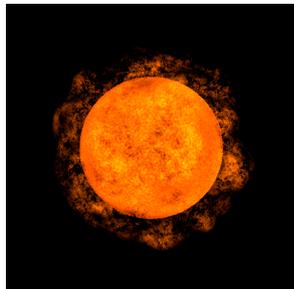
a. Star sphere



b. Surface shader



c. Prominence



d. Prominence with displacement

**Figure 4: Combining various shaders to create more realistic looking stars.**

The galaxy as a whole was imaged using subtle techniques that cannot be seen when looking at a single star. Every star in the simulation was bound to a transparent surrounding sphere with a radius 25 times larger than itself. The RenderMan® Shading Language's noise function was used to generate a believable regular pattern. The noise function was given the surface point of the sphere as a parameter and subtracting a constant value of 0.4 for all red, green blue (RGB) colors. The magnitude of the color was further determined by five octaves. Each of the

octaves enclosed the noise function where noise was multiplied by a value of 3.6 lacunarity. After color values were calculated for each surface point, opacity was defined using the values in Table 1.

	Red	Green	Blue
Opacity	0.02	0.02	0.03

**Table 1: Red, green, and blue opacity values**

Rendering of the movies was performed using RenderMan Pro Server 15.1 on iMac 9.1 machines with 2.66 GHz Intel Core 2 Duo processors, and 4 GB 1066 DDR3 memory running OS X version 10.6.4. RIB files were generated from Spiegel visualization system for input to PhotoRealistic RenderMan®. The RIB files were dispersed among 10 machines where PhotoRealistic RenderMan® was used to render the frames. A simple Bash shell script was used to feed a single RIB file at a time to reduce overhead and memory consumption.

#### 4. RESULTS

Rendering time for an average frame containing 14000 stars took approximately 4 to 6 minutes across workstations. The movies generated are on average 1 minute, 40 seconds in length at 25 frames per second. All frames had a 768 x 768 pixel resolution. Transitions between frames appear smooth but a careful viewer can sometimes perceive subtle discontinuities. The discontinuities only occur with slow moving stars at the edges of the simulation where they do not change positions enough between frames for the difference to appear. These transitions were hard to perceive and vary in severity depending on the input data, frame rate, and number of times interpolated between positions.

The effects of the final galaxy shader are most visible in the center of galaxy where many stars contribute to the final pixels. For a simulation of two super dense galaxies colliding with a larger galaxy we noted that as stars are pulled away from their home galaxies that this results in a visible reduction in the total brightness of their home galaxy mimicking what happens when real galaxies collide. Figure 5 shows a galaxy rendered using our approach.

#### 5. CONCLUSION AND FUTURE WORK

We found that PhotoRealistic RenderMan® could be used to generate realistic looking visualizations of galaxies and galactic events. Our technique takes advantage of noise algorithms, which mimic reoccurring patterns found in nature. This, combined with the fluid motion of the galaxy, can be used to create believable movies that are based on scientific data as opposed to an artist's depiction. Movie

quality appeared similar to that of the false color astronomical photos. More realistic images are possible with diligent shader design.

PhotoRealistic RenderMan®, while useful, does have limitations. When rendering large sets images we observed considerable slow down as size increased. Using a simple UNIX shell script we were able to reduce overhead and memory consumption by feeding one RIB file at a time to the renderer.

In the future we would like to visualize the effects created by the bending of light around black holes. However, RenderMan's ray-tracer is not set up to trace light along curves; it only traces light rays in straight lines from the camera and between objects. Another issue with this idea is that there is no explicit formula for the curves that the light follows around black holes, especially in the case where there are multiple black holes. In general the differential equations describing the shape of the space-time continuum cannot be solved analytically. The geodesics that the light rays follow need to be generated by integrating these equations numerically. Zink discusses an approach to general relativistic volume raytracing that accurately visualizes the matter around a single black hole [Zin08]. His approach involves generating geodesics by numerical integration. As part of our future work we plan to extend his technique to handle multiple black holes. Since the RenderMan® ray-tracer only works along straight lines it should be possible to make this algorithm work by dividing the curves into short straight line segments.

Finally, although RenderMan® has some limitations, it is a definitely a powerful and viable solution for visualizing scientific data. It readily provides a rich collection of shading and rendering techniques, with large libraries of shaders publicly available. A scientist with limited programming knowledge can easily modify them to suit their needs. RenderMan® should be used when appealing and realistic visualizations need to be created. However, for real time rendering it is not recommended. One final benefit to using RenderMan is that it runs on standard commodity hardware and can be distributed to multiple computers to take advantage of whatever a scientist has available.

## 6. REFERENCES

[Bis05] Hans-Peter Bischof and Jonathan Coles, A Movie Is Worth More Than a Million Data Points, Lecture Notes in Computer Science Publisher: Springer-Verlag GmbH, ISSN: 0302-9743 Subject: Computer Science Volume 3514/2005, Title: Computational Science ICCS 2005: 5th International Conference, Atlanta, GA.

[Esp10] Julio Espinal, Virginia Allen, Kwesi Amable, Reynold Bailey, and Hans-Peter Bischof, RenderMan's Power to Visualization's Rescue. 18th International Conference on Computer Graphics, Visualization and Computer Vision, 2010.

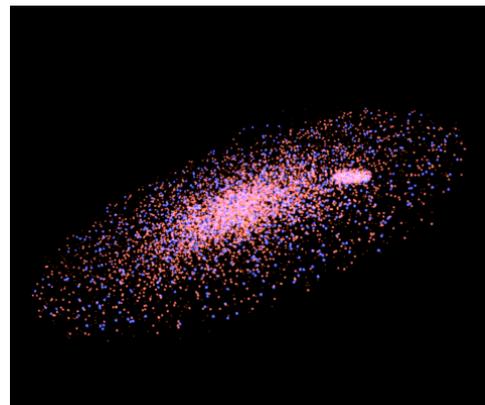
[Kau00] Kaufmann, M. (2000). Advanced RenderMan. San Diego, California: Academic Press. Nobel, S. (n.d.).

[Mol10] Karen Moltenbrey, Scientific Visualization, Computer Graphics World, Volume 33, Issue 7, (July 2010).

[Per85] Ken Perlin, An Image Synthesizer, Computer Graphics, Volume 19, Number 3, 1985.

[RIT10] Rochester Institute of Technology (2010). Retrieved October 20, 2010. Center for Computational Relativity and Gravitation, website: <http://ccrg.rit.edu/>.

[Zin08] Zink, B. (2008). Ray-tracing Black Holes. Saarbrucken, Germany, Germany: VDM Verlag Dr. Muller.



a. Galaxy shown without effect of galaxy shader.



b. Galaxy shown with effect of galaxy shader

**Figure 5: Images rendered with and without our galaxy rendering technique enabled. This particular dataset consisted of 14000 stars.**

# Use of virtual reality and constraint programming techniques in interactive 3D objects layout

KEFI Marouene

Laboratoire d'Ingénierie des  
Systèmes Automatisés  
Université d'Angers, 62 Avenue  
ND du Lac, Angers, France  
marouene.kefi@univ-angers.fr

RICHARD Paul

Laboratoire d'Ingénierie des  
Systèmes Automatisés  
Université d'Angers, 62 Avenue  
ND du Lac, Angers, France  
paul.richard@univ-angers.fr

BARICHARD Vincent

Laboratoire d'Etudes et de  
Recherche en Informatique  
Université d'Angers, 2 Bd  
Lavoisier, Angers, France  
vincent.barichard@univ-angers.fr

## ABSTRACT

In this paper, we propose innovative system in order to assist the user in a 3D objects layout context. Through a combination between virtual reality (VR) and constraint programming (CP) technique, user's 3D interaction and manipulation will be translated to incoming queries of a constraints solver which propagate constraints and generate a new possible solution. The computed solution is transmitted, as new positions of 3D objects, to virtual environment (VE) which reconfigures itself. We focus in this paper on the architecture of our system and we describe the implementation of several constraints and some first results.

## Keywords

Virtual environment, 3D interaction, Decision-making, 3D objects layout, constraint programming.

## 1. INTRODUCTION

A spatial configuration problem can be defined as a placement problem for which, and while satisfying the constraints, a positioning of components inside the container is looked for. It has applications in many industrial sectors. Often solved by hand from intuition and experience of designers, the development of automatic methods to solve the problem becomes a challenge at time when systems become more complex.

VR is defined as a system composed of software and hardware elements stimulating a realistic human interaction with virtual objects which are synthetic models of real or imaginary objects. The 3D interaction is the major component of VR, it allows the user to be able to change the course of events in a synthetic environment [Bowman, 1999].

VEs technology is now recognized as a powerful design tool in industrial sectors such as manufacturing, process engineering, construction, and aerospace industries [Zorriassatine et al., 2003].

However, in many cases, VEs are being used as a pure visualization tool for assessing the final design. VR can be used in many contexts of decision making involving several constraints, such as 3D objects layout which can be a tedious and costly task.

Thus the classic use of VEs does not provide assistance to the user in a 3D layout context and does not furnish indication on the best positioning of 3D objects. The integration of an intelligent module (constraints solver in our case) in VEs could resolve the interactive spatial configuration problem.

The notion of constraint is naturally present in several areas such as resources allocation, planning and industrial production. We can define a constraint as a property or condition that must be satisfied, it can be expressed as a relationship or a restriction on one or more variables.

To provide a solution of 3D objects layout problem, we present an intelligent virtual environment allowing the user to interact with virtual objects while respecting the predefined constraints of design. From a set of 3D objects, the user can select those which will constitute the 3D scene and specify their geometric properties (dimensions, colors ...) and semantic ones (temperature, light, vibration ...).

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.

## 2. RELATED WORK

Some works on the under constraints programs in VEs have been developed. For example, Xu et al., have treated the combination of physics, semantics, and placement constraints and how it permits to quickly and easily layout a scene [Xu et al., 2002]. The author generalized distributions and a richer set of semantic information leading to a new modeling technique where users can create scenes by specifying the number and distribution of each class of object to be included in the scene. Sanchez et al. have presented a general-purposed constraint-based system for non-isothetic 3D-object layout built on a genetic algorithm [Sanchez et al., 2002]. This system is able to process a complex set of constraints, including geometric and pseudo-physics ones.

More recently, Calderon et al., have presented a novel framework for the use of VEs in interactive problem solving [Calderon et al., 2003]. This framework extends visualization to serve as a natural interface for the exploration of configuration space and enables the implementation of reactive VEs.

It must be noted that these previous works are based on CLP and Prolog [Diaz and Codognet, 2001] or genetic algorithms. However, in the last few years, powerful CP-based solvers such as Gecode [Schulte, 1997] have been developed.

In spite of interest of previous works, they present some limits and can be extended in different directions. For instance, we envisage offering more interactivity (by using haptic feedbacks and stereoscopic images) to the user for more efficient object manipulation. In addition, and for more clarity, an explanatory information module will also be provided to justify the infeasibility of certain configurations proposed by the user.

## 3. SYSTEM DESCRIPTION

Our system is a 3D real-time environment based on CP techniques. It supports the resolution of interactive 3D objects layout. Through permanent communication, the choice of objects and constraints as well as user's 3D manipulation will be converted to queries sent to the solver. The work of the solver will be translated into automatic reconfiguration of VE (Figure 1). In addition, this system can present to the user many solutions (feasible spatial configurations) that will be able to explore by a specific device.

In order to intensify the user's immersion in the VE, a human-scale virtual reality platform is used in our first tests (shown in Figure 2 and described in the next section).

## 3.1 Architecture of interaction model

The aim of the interaction model is to make the correspondence between user's interactions with VE and inputs / outputs of the solver.

In our case, the work of the solver is based on a specific logic, depending on which, it is triggered by the addition of new constraints and it produces results in the form of new positions of objects. Thus, two aspects are concerned: (1) how the solver can respond to user's actions? (2) how the solver's results will interactively modify the VE?.

From a configuration of objects showed in the VE, the user can interact with it by moving some objects. This manipulation generates an event that will be used by the communications module (based on *threads*) to create new queries to the solver.

Acting according to these queries, the solver will produce new results sent directly to the virtual environment in order to update the current spatial configuration. Consider the simple example from an initial solution computed by the solver, the user moves the gray object (circled object) to the right (Figure 1). An event will be automatically generated from which the communication module "post" new constraints in the solver. These constraints will be applied on object which index is encapsulated in the event sent to the solver. Thus the solver will be recalled and the new position of the concerned object, and possibly those of other objects, will be encapsulated in another event sent to the VE (via the communication module) that extract new positions and reconfigure itself.

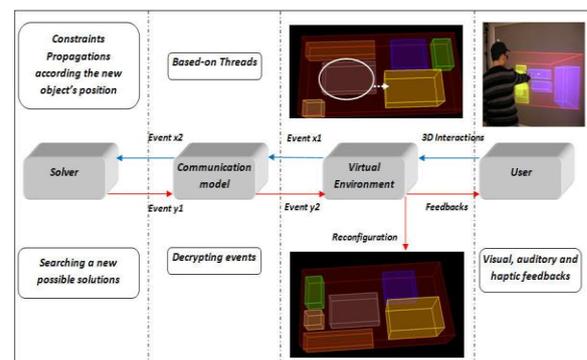


Figure 1. Architecture of intelligent VE.

## 4. VIRTUAL REALITY PLATFORM

The In order to intensify the user's immersion in the virtual world and assist him in his 3D arrangement, a human-scale virtual reality platform is used in our first tests (Figure 2). VIREPSE is a human-scale VE that provides force feedback using the SPIDAR

system (Space Interface Device for Artificial Reality) [Richard et al., 2006]. Stereoscopic images are displayed on a rear-projected large screen (2m x 2.5m) and are viewed using polarized glasses. Four motors are placed on the corners of a cubic frame surrounding the user. By controlling the tension of each string, the system generates appropriate forces (Figure 2).

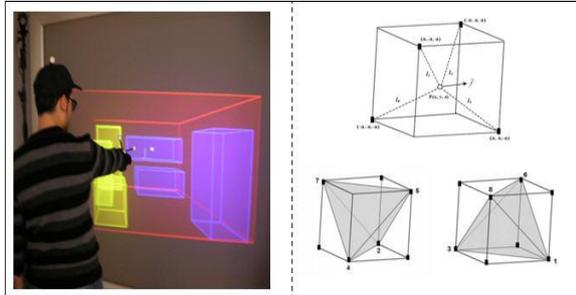


Figure 2. Multi-modal platform: VIREPSE

## 5. COMBINATION RV-CP

As mentioned before, our objective is to propose and implement interactive approach to solving interactive 3D layout problems [Kefi et al., 2010]. This approach is based on the architecture and interaction framework described above. From a GUI (Graphical User Interface), the user begins by selecting 3D objects to place and constraints to satisfy. Then the system will launch a dialogue with the solver to check the feasibility of the 3D arrangement. As illustrated in the next figure, the user can interact with the proposed solution (computed by the solver) by moving its constituent objects. After each displacement, the solver is recalled to consider new constraints and calculate new solutions or cancel the last displacement (if at least one constraint is violated). Once the new solution computed, the 3D environment is informed of the new positions of objects and will automatically reconfigure itself (Figure 3).

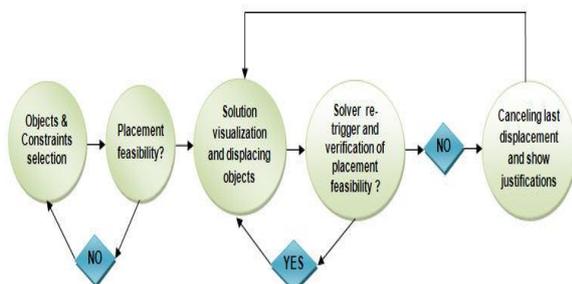


Figure 3. Illustration of the interactive approach

## 6. CONSTRAINTS IMPLEMENTATION AND FIRST RUN EXAPLES

In the case of our problem, constraints of arrangement can be divided into two categories: (1) Geometric constraints related to the physical placement of 3D objects. For example, the constraint of *no\_overlapping*, constraint *minimal\_distance* (the concerned object is away from other objects at least distance *dmin*). (2) Semantic constraints: from the fact that each object has a list of semantic attributes (light, temperature, vibration), this type of constraints uses these attributes to define the location of objects. For example, the temperature constraint uses the attribute temperature of the concerned object to place it away from sources of heat. It should be noted that several geometric constraints have been implemented allowing a first validation of our approaches. The implementation of semantic constraints is underway. The next part will be devoted to describe the firsts results obtained in order to validate our constraints implementation. It must be noted that we use the same propagation techniques for all the constraints. For each one, we use the same heuristics to select variables and their associated values. In this paper we present only one constraint: the *minimal\\_distance*.

### Minimum-distance-constraint

This constraint forces involved objects (cows in this example) to be far-off by a distance greater than or equal to a distance (*dmin*) specified by the user. In addition this constraint can be useful for example to put an object away from sources of heats. As shown in the following figure, objects can be placed in the space while keeping a minimum distance of *dmin*.

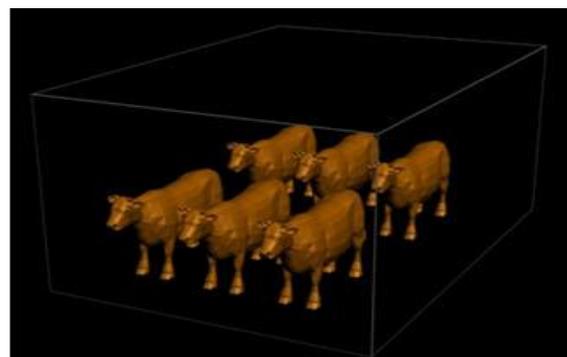


Figure 4. Illustration of the Minimum\_distance\_constraint

## 6.1 Response time

The response time is determined by the speed at which solutions are computed. Certain operations like displacing some expensive 3D objects and loading some big ones could slow down the 3D layout manipulations. So, interaction time cycle is depend not only on using the suitable technologies to model and implement constraint but also on the overall system architecture and interaction framework.

In order to evaluate the response time of our system, we have carry out some experiments with different number of objects and using only two kinds of constraints: minimum-distance and on\_floor constraints. The following figure shows the response time as a function of the number of 3D objects.

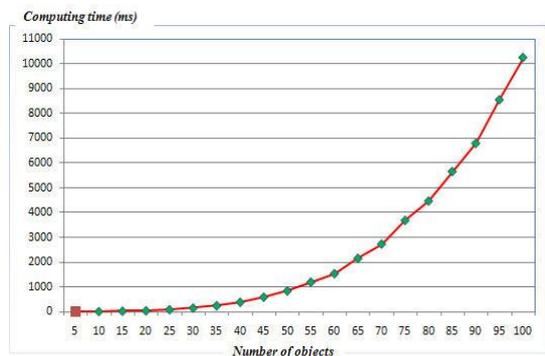


Figure 5. Computing time=f(number of objects)

## 6.2 Discussion

We can remark that the execution time increases with the number of objects to layout. Computing times obtained are sufficient to solve a real layout problem where the number of objects does not generally exceed fifty objects. Our current system can solve such problems in less than one second, which ensures real-time interaction and a response almost instantly.

## 7. CONCLUSION

We have presented an efficient system for interactive 3D objects layout problem solving. Based on a combination RV-CP, this system can be used to assist the user during a 3D configuration task (spatial configuration). Through a GUI, the user can add 3D objects to the environment and choose constraints for each. 3D manipulations and user interaction will be converted to new queries sent, through a structured communication module, to the constraint solver. Thus each moving objects retriggers the solver which, after propagations of constraints, is looking for new solutions and transmit it to the VE. In the future, some tracks are envisaged: increase realism and immersion through

the use of advanced 3D interaction techniques, increasing the size of the problem (increasing the number of objects to arrange).

## REFERENCES

- [Bowman, 1999] Bowman, D. (1999). Interaction Techniques for Common Tasks in Immersive Virtual Environments: Design, Evaluation, and Application. PhD thesis, Georgia Institute of Technology.
- [Calderon et al., 2003] Calderon, C., Cavazza, M., and Diaz, D. (2003). A new approach to the interactive resolution of configuration problems in virtual environments. Lecture notes in computer science, 2733 :112 – 122.
- [Diaz and Codognet, 2001] Diaz, D. and Codognet, P. (2001). Design and implementation of the gnu prolog system. Journal of Functional and Logic Programming, Vol. 2001, No 6.
- [Kefi et al., 2010] Kefi, M., Richard, P., and Barichard, V. (2010). Interactive configuration of restricted spaces using virtual reality and constraints programming techniques. In International Conference on Computer Graphics Theory and Applications.
- [Richard et al., 2006] Richard, P., Chamaret, D., Inglese, F., Lucidarme, P., and Ferrier, J. (2006). Human-scale virtual environment for product design : Effect of sensory substitution. The International Journal of Virtual Reality.
- [Sanchez et al., 2002] Sanchez, S., Roux, O. L., Inglese, F., Luga, H., and Gaildart, V. (2002). Constraint-based 3dobject layout using a genetic algorithm.
- [Schulte, 1997] Schulte, C. (1997). Oz explorer : A visual constraint programming tool. Proceedings of the Fourteenth International Conference on Logic Programming, pages 286–300.
- [Xu et al., 2002] Xu, K., Stewart, J., and Fiume, E. (2002). Constraint-based automatic placement for scene composition. In Graphics Interface Proceedings, University of Calgary.
- [Zorriassatine et al., 2003] Zorriassatine, F., Wykses, C., Parkin, R., and Gindy, N. (2003). A survey of virtual prototyping techniques for mechanical product development. Journal of Engineering Manufacture, Part B :217.