The 15th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2007

in co-operation with

EUROGRAPHICS

W S C G ' 2007

University of West Bohemia
Plzen
Czech Republic

January 29 – February 1, 2007

Posters proceedings

Co-Chairs

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

Edited by
Vaclav Skala

WSCG'2007 Posters Proceedings

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

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

Printed at the University of West Bohemia

Hardcopy: *ISBN 978-80-86943-99-2*

WSCG 2007

International Programme Committee

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

WSCG 2007 Board of Reviewers

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

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

Contents

Paper Code	Title	Page
A07	Li,B.-B., Zhao,Z.-H., Geng,Z.-F.: POSTER: An Integrated Method for 3-D Textile Simulation (China)	1
B37	Prochazkova,J., Prochazka,D.: POSTER: An Implementation of NURBS Curve Derivatives in Engineering Practice (Czech Republic)	5
B97	Lapin,K.M., Rakovskaja,O.: POSTER: Combining the Interaction Styles to Display Complex Data in Decision-making System (Lithuania)	9
C29	Hast,A: POSTER: Topology Maps for Anisotropic Shading (Sweden)	13
D02	Gage,P., Semwal,S.K.: POSTER: Accelerated Pan and Zoom for Ray Traced Animation (United States)	17
D13	Petrícek,M., Pelikán,J., Horák,M.: POSTER: Duct Tracking in 3D Medical Data (Czech Republic)	21
D47	Silva,N.R., Manssour,I.H.: POSTER: A Curvilinear Cutting Tool for the Analysis of Oceanography Data (Brazil)	25
D83	Rachkovskaya G., Kharabayev Yu., Rachkovskaya N.: POSTER: Computer Composition of Transformed Classical Surfaces as the Ways and Means for Constructing Visual Models of Realistic Objects (Russia)	29
D89	Kobori,K., Hori,T., Nakamura,N., Nishio,K.: POSTER: Development of Virtual Goldfish Scooping Using Haptic Device (Japan)	33
F02	Preuss,S., Schmitt,A.: POSTER: Design Patterns for Multithreaded Software Pipelines in Real-Time Applications (Germany)	27
F59	Saitoh,M., Miyamoto,K.: POSTER: Controlling 3D Characters by Hand Motion (Japan)	41
G37	Nishio,K., Iwaizumi,T., Nakamura,N., Kobori,K.: POSTER : A Method for Generating Developments using Decomposition into the Meaningful Components of 3D Polygon Models (Japan)	45
H43	Dodo,M., Torguet,P., Sibilla,M., Jessel,J.P.: POSTER: Customizable Tool for Complex Systems Visualization (France)	49
H73	Oh,M.-J., Hur,S., Kim,T.-W.: POSTER: Finding Direction of Intersection Curve in Critical Cases of Surface-surface Intersection (Korea)	53
H79	Kolcun,A., Pravda,D.: POSTER: Isosurface Extraction from Volumetric Data (Czech Republic)	57

An Improved BRDF Method for 3D Textile Simulation

Bei-Bei Li
College of Information Sience and Technology, Donghua University, 1882 Yan/an Xilu Road, China, 200051, Shanghai
lbb@mail.dhu.edu.cn

Zhi-Hong Zhao
College of Information Sience and Technology, Donghua University, 1882 Yan/an Xilu Road, China, 200051, Shanghai
zhaozh@mail.dhu.edu.cn

Zhao-Feng Geng
College of Information Sience and Technology, Donghua University, 1882 Yan/an Xilu Road, China, 200051, Shanghai
gengzf@dhu.edu.cn

ABSTRACT

There are some methods suggested by researchers were used to simulate one or two characteristics of textile in the past few years. With 3D geometry models, most of these methods are somewhat complex, and it is difficult to simulate fabric attributes. This paper represents a method for textile simulation based on improved BRDF illumination models. As one simple type of BRDF model, Phong illumination model can be used to simulate fabric 3D geometry model. The more complex BRDF illumination model may be used to simulate fabric 3D structure. The pixel color values, derived from simulating fabric 3D structure, is taken as the parameters to be entered into Phong illumination model. And thus both the fabric 3D geometry model and structure can be achieved, based on which the textile can be simulated successfully and obtain a perfect display effect.

Keywords

3D textile, BRDF model, illumination model, simulation

1. INTRODUCTION

The representation of 3D textile model is one of the important techniques in the fashion CAD. The structure of 3D textile was one of research emphases. With the development of computer technology and the improvement of simulation for textile, more and more researchers use computers to represent 3D textile structure. There are some methods to simulate textile, such as Phong illumination model [1] and BRDF model [2].

Phong illumination model and BRDF model are popular techniques in computer graphic field. More specifically, the former is the simplest type of the latter.

Phong illumination model is suitable for simulating those materials with smooth surface.

It has been improved by some researchers [1] by means of controlling some parameters, such as the proportion coefficients of circumstance, diffuse and mirror reflection. They calculated the values of the angle of light source incidence and that between the mirror reflection and the sight direction according to the structure of the yarn and woofs [3].

BRDF model was widely applied to simulate material with coarse surface. It can produce the effect of microscopic features shadowing certain directions or reflecting light to another specific direction [4].The models, based on BRDF model, could be used to simulate textile microscopic feature in the plane surface without winkles. These models cannot be combined with the 3D textile geometric model.

In this paper, the improved BRDF model is suggested to describe 3D textile structure while the Phong illumination model is used to represent 3D geometry illumination effect. The integration of these two kinds of models is done by method of synthesis. The pixel color values, derived from simulating fabric 3D structure, is taken as the parameters to be entered into Phong illumination model.

According to the textile material and 3D geometry model, we can adjust its structure to achieve a good simulation effect.

2. BRDF MODEL

In the computer graphic field, BRDF models are four-dimensional functions, which can be used to describe the reflection distribution at a surface point depending on incoming and outgoing light directions. There are various strategies for modeling BRDF. This paper implements Schlick's BRDF Model, which is simple and easy to realize. It can be represented as

$$f = \frac{1 - G(u)G(v)}{\pi} + \frac{G(u)G(v)}{4\pi uv}\left(\frac{\partial}{(1+\partial t^2 - t^2)^2}\right) \quad (1)$$

Where,

t, u and v are the cosines of α, θ and θ' (shown in Fig.1),

$G(u)$ and $G(v)$ can be defined as in Eq.(2), f is the reflection intensity received by observers.

$$G(x) = \frac{x}{\partial - \partial x + x} \tag{2}$$

Where ∂ ($0 \le \partial \le 1$), shown as in Eq.(2), is the roughness of the surface. If $\partial = 0$, there is no diffuse but mirror reflection on the surface. And if $\partial = 1$, only diffuse reflection exists.

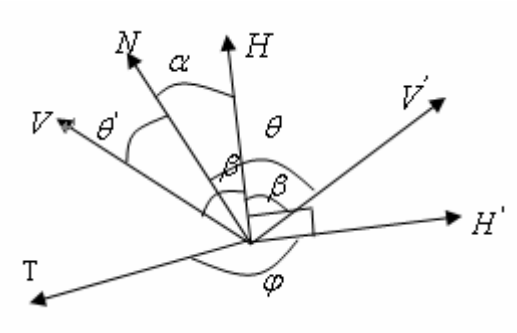


Figure 1. BRDF model.

In Fig.1,

V is the reflect light,

V' is the incident light,

N is the surface normal,

H is the internal bisector of incident light and reflex light,

α is the angle between H and N,

θ is the angle between V' and N,

θ' is the angle between V and N.

We make some improvement on this BRDF model for its disadvantages of complex modeling, large numbers of parameters, endless of calculations and poor real-time ability.

3. INTEGRATED MODEL IN THIS PAPER

In this Sector, Phong and BRDF Illumination model are made some improvement. The new model used in this system can be done by integration of the two improved models.

Improvement of Phong illumination model

This paper improves Phong illumination model based on the characteristics of 3D fabric model. According to the textile material simulated, the values of parameters can be defined as follows:

$$k_a = 0.1, k_d = 0.7, k_s = 0.3, n = 5, I_{pa} = 0.1 I_{pd}.$$

Where I_{pd} is color value of the simulated point. We assume I_{pd} =1. As there is only one light source in the model, then Eq.(1) can be turned into

$$I = 0.01 I_{pd} + 0.7 I_{pd} \cos i + 0.3 \cos^5 \theta \tag{3}$$

The improved Phong Illumination model can be used to represent the textile 3D geometry illumination effect. It can be integrated with the following improved BRDF model used to simulate textile structure by the method of integration.

Improvement of BRDF model

The process is used to imitate textile structure. In order to imitate textile more vividly and efficiently, we make some improvement on the BRDF model.

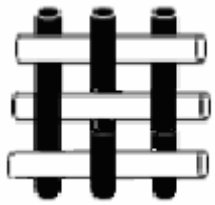


Figure 2. Structure of textile.

The textile structure in this paper is shown in Fig.2.

Where, the relationship of Warp and woof can be obtained, shown as in Fig.3.

Figure 3. The fluctuation relationship of yarns.

As a result, we can get

$$f(w) = \sin(lw/360) \tag{4}$$

Where $lw/360$ varies from $(2k+1)\pi$ to

 ISBN 978-80-86943-99-2

$2k\pi$ (k =0,1,2,...30), l is the distance between two adjacent yarns and w is that between the point imitated and the initial state. The initial value of $f(w)$ is assumed as 0.

Here, if $f(w)$ of Eq.(4) times f of Eq.(1), we will obtain

$$I = f(w)f \tag{5}$$

Where I is reflection intensity received by observers.

Integration of Phong illumination model and BRDF model

The image generated from the improved BRDF model can be used as the texture. We use integration to obtain the integration of the two improved models presented above. The simulation effect can be shown efficiently, from which we can see the textile structure, bumps and winkles clearly.

The illumination model of the 3D structure may be got from Eq.(6). Textile structure light intensity value, generated from BRDF model, is defined as I_{ph}. And the formula (3) can be obtained

$$\mathrm{I} = 0.1 \times 0.1 I_{pd} \times I_{ph} + 0.7 I_{pd} \times I_{ph} \cos i + 0.3 \times 0.1 \cos^5 2i \tag{6}$$

Where I_{ph} is the structure color value of each pixel.

4. SIMULATION RESULT

The simulation is completed on a single 2.4GHz CPU of a PC system. In this 3D fabric simulating system，it reduces the simulation time to less than 0.1second. The result is vivid as following Fig.4 to 6:

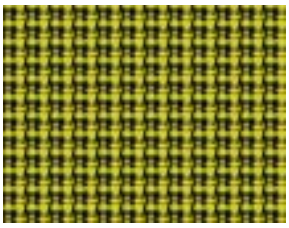


Figure 4. Textile structure imitated in BRDF model.

Figure 5. 3D model imitated in Phong illumination model.

Figure 6. Integration of BRDF model and Phong illumination model.

Fig.4 shows the textile structure simulated by the improved BRDF model. It is used as the texture pattern in the process of integration.

We can see the simulation effect of textile geometry model by improved Phong Illumination model. It is used as the curved surface in the process of integration.

Fig.6 shows the effect of integration, from which we can see the textile structure, bumps and winkles clearly.

In computer simulation and animation, depending on the simulation speed, only 3D geometry model can be simulated if the Phong Illumination model is adopted while only the 3D structure can be simulated if the BRDF model is applied.

In this method, textile 3D geometry model and structure can be simulated on the base of computer simulation speed considered.

5. CONCLUSION

The imitation of 3D textile is a complex system, covering many kinds of technologies, and it is of great importance in many fields, such as fabric designing, movie making, animation studios, virtual reality, etc.

In this paper, we propose a 3D textile imitation algorithm based on the two improved models of BRDF and Phong. It is an efficient method for the real-time simulation for fabric features. We first simulate the textile microscopic features using improved BRDF model. According to the material and structure, parameters can be controlled to simulate microscopic feature. The improved Phong illumination model is then used to simulate the textile macroscopic geometry features, such as bumps and winkles. We use integration to integrate the display effect of the micro- and macroscopic features mentioned above. Experiments demonstrate that our method is efficient and effective. As the arithmetic

avoids recursive calculation, the simulation has a fast speed. The algorithm proposed will contribute to the 3D textile design greatly. Future work will therefore concentrate on the real-time dynamic simulation for textile in our system.

6. ACKNOWLEDGMENTS

I'd like to thank Miss Min Xu, one of my best friends for her help.

7. REFERENCES

[1] Phong B.T. Illumination for computer generated pictures. *Comm. of the ACM*, 18(6) (1995)449-455.

[2] Ashikhmin, M., Premoze, S., Shirley, P. A microfacet-based BRDF generator, *in Proceedings of SIGGRAPH*,(2000) 65-74.

[3] Yan-Gang Feng, YangBaoZhu. The fabric illumination model and its computer imitation. *ZheJiang Engineering Institute transactionv*, 16(3)(1999)198-200.

[4] C. Schlick. An inexpensive brdf model for physically-based rendering. *Computer Graphics Forum (Proceedings of Eurographics'94*), 13(3) (1994)233-246.

[5] TangRongXi. *Computer Graphic tutorial*. Science publishing company,2000,BeiJing China.

[6] Catmull, E. A Subdivision Algorithm for Computer Display of Curved Surfaces, Ph.D. dissertation, University of Utah,1974.

[7] David F. Rogers. *Procedural Elements for Computer Graphics*. China Machine Press, 2002, BeiJing China.

Implementation of NURBS Curve Derivatives in Engineering Practice

Jana Procházková
Faculty of Mechanical Engineering
Brno University of Technology
616 69 Brno, Czech Republic
jprochazkova@gmail.com

David Procházka
Faculty of Business and Ec.
Mendel University in Brno
613 00 Brno, Czech Republic
prochazka@pef.mendelu.cz

ABSTRACT

Derivatives are a very important tool of computations in an engineering practice on the graphics structures. B-Spline functions are defined recursively, so the direct computation is very difficult. In this article new direct proof of the formula used for simpler direct computation is shown. The paper also presents own method of programming derivatives of NURBS curves by means of the proven formula which is implemented in German engineering software RFEM 3D.

Keywords: NURBS, derivative.

1. INTRODUCTION

NURBS (Non-Uniform Rational B-Splines) are used in graphic software to design free-form objects and also in various engineering software for designing buildings, bridges, etc. Derivatives are necessary in these applications for physical computations such as stress analysis.

The first part of this paper introducing a new direct proof of the formula used for computing derivatives of B-Spline functions. The gist of this proof is that we don't need any other characteristics of B-Spline function as in the case of the proof in [PT92]. Therefore our proof is more straightforward and clear. Second part of paper presents our method of programming derivatives of the NURBS. We deal with verifying of the continuity with respect to a special case and all other issues which are necessary for succesfull implemenation of the derivatives.

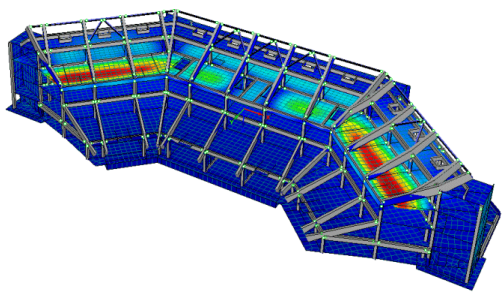


Figure 1: Stress Analysis in RFEM3D – Ice Hockey Stadium

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

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

2. RELATED WORK

Our work with NURBS object has started by studying and implementation of curves, general surfaces (see [Pro06a]) and basic types of NURBS solids – cone, cylinder, sphere, cube (see [Pro06b]) to RFEM 3D CAD system. As mentioned, the derivatives are the key for computing stress analysis, therefore we started analyze them. First conception is published in [Pro05].

The theoretical background for this paper is mostly described in three sources. First one is a book by Peigl and Tiller [PT92] about NURBS objects. This book contains the algorithms of computing points on NURBS curve and reveal the proof of derivatives. Second source is a Floater's paper [Flo92b] about derivatives of Bézier curves. In third one [Flo92a] two ways of the first derivative of rational B-Spline curve in terms of its control points and weights are described.

3. B-SPLINE AND NURBS

This section briefly outlines the theory about B-Spline and NURBS used in the rest of paper. B-Spline (NURBS) are defined with B-Spline functions recursively. Let $t = (t_0, t_1, \ldots, t_n)$ be a knot vector (non-decreasing sequence of real numbers). Then B-Spline function is defined by:

$$N_i^0(t) = \begin{cases} 1 & \text{for} \quad t \in \langle t_i, t_i + 1) \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

$$N_i^k(t) = \frac{t - t_i}{t_{i+k} - t_i} N_i^{k-1}(t) + \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} N_{i+1}^{k-1}(t),$$

where $0 \le i \le n-k-1, 1 \le k \le n-1, \frac{0}{0} := 0$.

We added the control points $P_0, P_1, \ldots, P_m$ ($P_i \in \mathbf{R}^d$) and the knot vector $\mathbf{t} = (t_0, t_1, \ldots t_{m+n+1})$. Then B-Spline curve is defined:

$$C(t) = \sum_{i=0}^{m} P_i N_i^n(t) \tag{2}$$

Let $w_0, \ldots, w_m$ be a vector of weights, which are attached to every control points. Weights (non-negative numbers) determine the influence of its control point. So, NURBS curve of degree n is defined by:

$$C(t) = \frac{\sum_{i=0}^{m} w_i P_i N_i^n(t)}{\sum_{i=0}^{m} w_i N_i^n(t)} \tag{3}$$

where $t \in \langle t_n, t_{m+1} \rangle$.

4. PROOF OF THE DERIVATIVE FORMULA

Let us show the new proof of the formula for computations of NURBS curve derivatives.

Theorem 1. *Let $C(t)$ be the B-Spline curve from previous section. Then its first derivative is evaluated by:*

$$C(t)' = \sum_{i=0}^{m} N_i^n(t)' P_i, \tag{4}$$

where

$$N_i^n(t)' = \frac{n}{t_{i+n} - t_i} N_i^{n-1}(t) - \frac{n}{t_{i+n+1} - t_{i+1}} N_{i+1}^{n-1}(t).$$

Proof: The proof will be done by complete induction with respect to n.
1. $n = 0$
Obviously, the derivative is equal to zero in all cases. The theorem holds for $n = 0$.
2. Let us suppose, that the formula holds for $k = 0, 1, \ldots, n$. The idea of proof is in comparing different expressions of $N_i^{n+1}(t)'$. The first is done by computation of the expression:

$$N_i^{n+1}(t)' =$$

$$\left(\frac{t - t_i}{t_{i+n+1} - t_i} N_i^n(t) + \frac{t_{i+n+2} - t}{t_{i+n+2} - t_{i+1}} N_{i+1}^n(t) \right)'.$$

According to the induction suppose, we know the expressions of degree n: $N_i^n(t)'$ and $N_{i+1}^n(t)'$ and we substitute them. The second expression is from Theorem 1 for degree $n+1$. We get these two summands:

$$\begin{aligned}
N_i^{n+1}(t)' &= \frac{1}{t_{i+n+1} - t_i} N_i^n(t) - \frac{1}{t_{i+n+2} - t_{i+1}} N_{i+1}^n(t) \\
&+ \frac{n}{t_{i+n+1} - t_i} \left(\frac{t - t_i}{t_{i+n} - t_i} N_i^{n-1}(t) \right) \\
&- \frac{n}{t_{i+n+2} - t_{i+1}} \left(\frac{t_{i+n+2} - t}{t_{i+n+2} - t_{i+2}} N_{i+2}^{n-1}(t) \right) \\
&+ \frac{n}{t_{i+n+1} - t_i} \left(\frac{t_{i+n+1} - t}{t_{i+n+1} - t_{i+1}} N_{i+1}^{n-1}(t) \right) \\
&- \frac{n}{t_{i+n+2} - t_{i+1}} \left(\frac{t - t_{i+1}}{t_{i+n+1} - t_{i+1}} N_{i+1}^{n-1}(t) \right)
\end{aligned}$$

and

$$\begin{aligned}
N_i^{n+1}(t)' &= \frac{1}{t_{i+n+1} - t_i} N_i^n(t) - \frac{1}{t_{i+n+2} - t_{i+1}} N_{i+1}^n(t) \\
&+ \frac{t - t_i}{t_{i+n+1} - t_i} \left(\frac{n}{t_{i+n} - t_i} N_i^{n-1}(t) \right) \\
&- \frac{t_{i+n+2} - t}{t_{i+n+2} - t_{i+1}} \left(\frac{n}{t_{i+n+2} - t_{i+2}} N_{i+2}^{n-1}(t) \right) \\
&- \frac{t - t_i}{t_{i+n+1} - t_i} \left(\frac{n}{t_{i+n+1} - t_{i+1}} N_{i+1}^{n-1}(t) \right) \\
&+ \frac{t_{i+n+2} - t}{t_{i+n+2} - t_{i+1}} \left(\frac{n}{t_{i+n+1} - t_{i+1}} N_{i+1}^{n-1}(t) \right)
\end{aligned}$$

First three parts are identical. The equality is not evident for the last two expressions. Multinomial nN_{i+1}^{n-1} is exclude and we can make a special step – in the numerator we add and subtract $t_{i+1} t_{i+n+1}$. Then we got the desired equality. This finishes the proof and implies:

Corollary 1. *Let $C(t)$ be a NURBS curve defined in previous section. Then its derivative is of the form:*

$$C(t)' = \left(\frac{\sum_{i=0}^{m} w_i P_i N_i^n(t)}{\sum_{i=0}^{m} w_i N_i^n(t)} \right)' = \tag{5}$$

$$= \frac{\sum_{i=0}^{m} w_i P_i N_i^n(t)' \sum_{i=0}^{m} w_i N_i^n(t)}{\left(\sum_{i=0}^{m} w_i N_i^n(t)\right)^2}$$

$$- \frac{\sum_{i=0}^{m} w_i P_i N_i^n(t) \sum_{i=0}^{m} w_i N_i^n(t)'}{\left(\sum_{i=0}^{m} w_i N_i^n(t)\right)^2}.$$

It's proven, that derivative formula holds and can be used in engineering computations without any limitations.

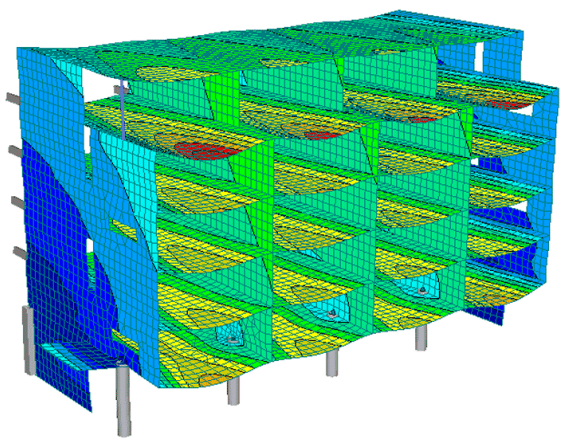


Figure 2: Analysis of Stress – Building

5. IMPLEMENTATION OF DERIVATIVES

The algorithm has two parts. The first one is the verification of continuity of $C(t)$ for particular input value of parameter t, the second one contains the computations.

 ISBN 978-80-86943-99-2

Verification of Continuity

Let $C(t)$ be a NURBS curve of degree n defined in theoretical part of article. First, we have to define the multiplicity of one knot in the knot vector. The value $\bar{t}$ has multiplicity s, when knot vector $\mathbf{t}$ contains value $\bar{t}$ s-times. Then, $C(t)$ is C_{n-s} continuous at the point $C(\bar{t})$ if the derivative of $(n-s)$-th order in $C(t)$ exists.
Let us now summarize the verification of curve continuity.

1) We find the interval of the knot vector with $\bar{t} \in \langle t_i, t_{i+1})$.

2) We find the multiplicity of $\bar{t}$.

3) We compute the continuity in the point $C(\bar{t})$.

4) We treat the special cases.

Special Case

Some curves have a points, which continuity is C_0 but the tangent and derivatives exist here. Generally:
Let have B-Spline curve $C(t)$ with arbitrary control point $P_i = C(\bar{t})$. $\bar{t}$ has multiplicity equal to degree of curve, then the continuity in point P_i is C_0. But if the control points P_{i-1}, P_i, P_{i+1} lie in the line then the derivative in the middle one exists.

Example of this case should be a circle. Figure 3 shows such circle as NURBS. For the parameter $t = 0.25$ the continuity in the point P_2 equals zero. However points P_1, P_2, P_3 lie in the line then the derivative in P_2 exists – it is line P_1P_3.

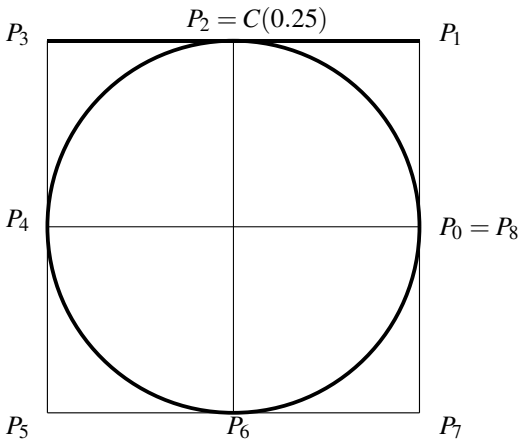


Figure 3: NURBS circle, degree = 2, knot vector $(0,0,0,0.25,0.25,0.5,0.5,0.75,0.75,1,1,1)$

Computation

The computation has two parts: initialization and computational algorithm.

a) Inicialization

- Inicialize basic functions N_i^n.

```
if t in <t_i, t_i+1)
  N[0][i] = 1;
  N[0][j] = 0; (j != i)
```

- Make projectivization of the control points (i.e. connecting the points with their weights).

```
for (i=0; i<array_length; i++)
  Point4d[i][0] =
  weights[i]*ControlPoint[i];
```

b) Computation

It is necessary to divide the computation of formula (5) in Corollary 1 into several parts.

Multinominal $N_i^j(t), j = 1,..,n$, where n is the degree of the curve, is computed by deBoor algorithm (see [Bo76], [PT92]).
Multinominal $N_i^n(t)'$ is computed by use of the proven formula in previous section means of N_i^{n-1} (deBoor algorithm for degree $n-1$).

Let us remark that the zero denominator is treated separately by defining $0/0 = 0$ which is typical for NURBS.

6. DERIVATIVES ON SURFACE

B-Spline (NURBS) surface is tensor product of two B-Spline (NURBS) curves with its parameters. Computing of partial derivatives is analogous to the curves. We obtain these derivatives by computing derivatives of basis functions N_i^p and N_j^q.

$$\frac{\partial^{k+s}}{\partial^k u \partial^s v} S(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}^{(k)} N_{j,q}^{(s)} P_{ij} \qquad (6)$$

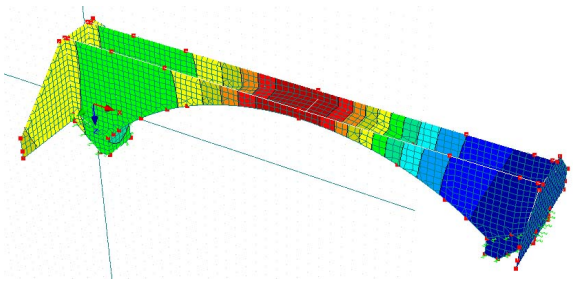


Figure 4: Analysis of Stress – Bridge

7. CONCLUSION

The derivatives of NURBS curves are in literature discussed only briefly and theoretically, therefore whole theory and its implementation is described in this article. Their importance in technical practice is enormous – physical calculations, building industry and so on.

The discussion in this article has given an overview of derivatives of B-Spline (NURBS). We demonstrated a new proof of known formula for computation of derivatives, which is more clear and straightforward. We also

reveal the possibility of programming with respect to the verification of the continuity and special case.

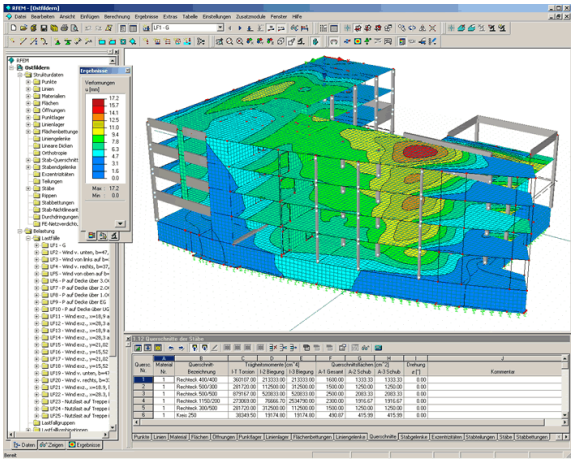


Figure 5: RFEM 3D

REFERENCES

[Bo76] de Boor, C. A Practical Guide to Splines. New York, 1976.

[Flo92a] Floater, M.S. Evaluation and properties of the derivatives of a NURBS curves. Mathematical methods in CAGD. pp. 261-274, 1992.

[Flo92b] Floater, M.S. Derivatives of rational Bezier curves. CAGD 9. pp. 161-174, 1992.

[PT92] Pegl, L. and Tiller, W. The NURBS Book. Springer-Verlag. 1992.

[Pro05] Procházková, J. Procházka, D. Derivatives of B-spline function. 25. Konference o pocitacové grafice. pp. 199-204, 2005.

[Pro06a] Procházková, J. Application of NURBS Surfaces in Engineering Software. Moderní matematické metody v inzenýrství (3MI). 2006.

[Pro06b] Procházková, J. Procházka, D. Implementation of Cylinder and Cone as NURBS Surfaces in Engineering Software. SCG 2006.

[RFEM] www.dlubal.com

 ISBN 978-80-86943-99-2

Combining the interaction styles to display complex data in decision-making system

Kristina Moroz Lapin, Olga Rakovskaja
Vilnius University
Naugarduko 24
LT-03225 Vilnius,
Lithuania

kristina.lapin@mif.vu.lt, olga.rakovskaja@mif.vu.lt

ABSTRACT

Analyzing the large amount of complex data is becoming increasingly difficult. This problem impels the interface designers to search for the methods to display such data in a usable way. Combining human computer interaction styles with information visualization techniques provides some solutions. Displaying the relevant data in appropriate level of abstraction can be a good interface design solution. The most important data is focused when the others are accessible but presented in a higher level of abstraction. Presenting the same data in a different manner when multiple views are used also can give good results. The combination of techniques and styles benefits the analyst' decision-making process.

Keywords

Human computer interaction, interaction styles, visualization techniques, focus+context, timeline, fisheye view, multiple views.

1. INTRODUCTION

The interface of credit insurance system should provide an effective way to support the analysis process of relevant information while granting the credit insurance limit. The credit insurer takes into account many interdependent data while making the decision of the possible credit limit for a particular insurant. The analysed data have temporal, structural and logical dependencies. The goal is to provide support for insurer making decision.

Current systems used by our customers lack usability. Users have difficulties viewing the dependencies that were provided by either the structural or linear views.

The first system presents the whole structure of related data in one window that clutters the display (see Figure 1). The analyst hardly finds the useful data.

The second – linear view – shows the related data in one row (see Figure 2). The screen does not contain all data. Some data is accessible by scrolling. Such presentation increases the chances of missing important 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.
Copyright UNION Agency – Science Press, Plzen, Czech Republic.

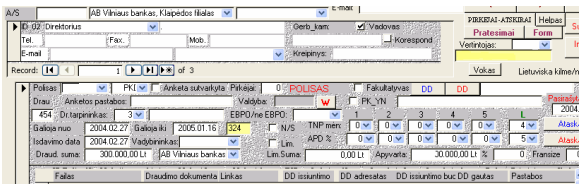


Figure 1. The whole structured record is presented on the one screen

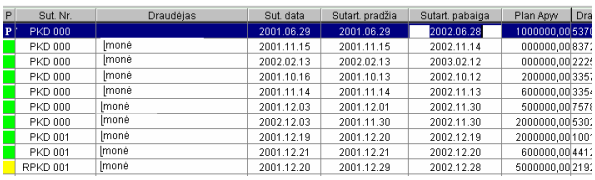


Figure 2. Linear view: the record is presented in one line.

In this paper we propose the combination of focus-context and timeline techniques applied in a multiple-view window to facilitate the analyst to make decisions.

2. PROCEDURE OF THE ASSESSMENT OF CREDIT LIMIT INSURANCE

Estimation of the credit limit is essential for the credit insurance service. While making the credit limit insurance decision the analyst gains the information

from the internal and external data sources. Internal data bases provide information of the previous insurance operations. External sources include financial indexes, such as status, turnover, sales, buying, the capital, spending, liquidity ratio, the development, and the events from the past contracts. Queries to internal sources are processed immediately. Queries to external sources block the decision-making process until all answers will be received.

Having the required information the analyst decides either to grant the credit limit or to deny the application. While granting the credit insurance and assessing credit limit, error prevention is essential. Errors appear when useful data is not shown and when imported data is not highlighted. Also they appear if relations between data are not clear and important information is mixed with less important.

Error prevention would be more effective when information with higher priority would be highlighted and visible on screen. Because of the large amount, the data should be shown in the different levels of abstraction, such as the overview of the relevant important data and easy accessible detailed view.

3. USABILITY DIMENSIONS

The definition of usability states that usability is effectiveness (accuracy and completeness with which users achieved task goals), efficiency (the task time users expended to achieve task goals) and subjective satisfaction (positive attitude to the use of the visualization) with which a specified set of users can achieve a specified set of tasks in a particular environment [ISO97a]. In order to promote usability three categories of usability principles are proposed: learnability, flexibility, and robustness [Dix03a].

These categories are subdivided into more specific principles. Insurer should see the data hierarchy relation according to relevant priorities. The points of interest should be visible in detailed view, related aspects only visible, but not detailed, until the user chooses them. Events should be organised in succession. So, the *timing scale* is important.

The large amount of data and complex relations requires effective usage the screen area. Therefore, *visibility* of the whole available information is significant. Essential subcategory of robustness is *error handling*. This action prevents users from committing a mistake or slip, or allow user to recover from erroneous path.

The task should be less time consuming comparing with the current system. So, the *efficiency* goal should also be raised for decision-support system.

4. VISUALIZATION AND INTERACTION APPROACHES

Visual data analysis requires both angles: overview and details-on-demand. First, insurer needs to overview the list of answered queries. In the overview user focuses on important aspects of the analyzed case. Then, user accesses details of the data.

Focus-context techniques allow having context information with the detail at the focus of attention. Focus-context approach allows easy move between views at different scales. The overviews of the four focus-context techniques – distortion, rapid zooming, elision and multiple windows – among others are given in [Spe00a]. The eight design guidelines [Bal00a] facilitate the solution of displaying information in multiple views.

Time and visualization can be considered from two sides: visualizing temporal values (such as data, periods, etc.) and ordering the other data on the time axis (such as monthly spending, weekly sales, etc.) [Dix00a]. Timeline is a technology that could be applied to visualize events in a time axis. Event, tasks and other information can be placed in parallel bars each with own objective. Bars are divided using colours or lines. This method both reduces the chances of missing information and streamlines access to details while remaining tailorable and easy transferable between tasks [Pla96a].

5. VISUALISATION OF COMPLEX INFORMATION

Complex information means the set of undivided objects that cannot be expressed by other single object. Complex relations set unambiguous relations between complex and single information objects. The case of credit limit insurance concerns several structured objects such as insurer, policy, claims and so on. All of them have relations between each other.

Decision making requires that complex data should be properly organised. Furthermore, analyzed data should be available in different views and scales.

Multiple-view approach is proposed because user needs to deal with related data in different levels of abstraction. The need for different views accords with the rule of diversity from [Bal00a]. This bifocal technique allows displaying the whole information space in one screen.

Rule of parsimony suggests using multiple views minimally [Bal00a]. The number of used visualisations is reduced to three. By the concept a screen is divided into three vertical parts. The biggest one displays detailed view of the chosen data, another two are for relating sets of the data. One of them contains the basic data of information user currently

studying. Another one is for displaying the results of the search.

The visualisations are synchronised. Selection in the left pane immediately updates the middle and right panes. Selection in the right pane adapts the middle pane view. These points are in harmony with the rules of self-evidence and consistency [Bal00a]. Such display's division allows using the whole workspace and shows the dependencies between the objects. Also some standard means (prints, colour, etc.) can be used to highlight the most important data.

Both parts that display the set of data may contain large amount of data. The fisheye view technique provides better visibility in case the large amount of the data is presented (see Figure 3). This method accords with the rule of space/time resources optimization that indicates to balance the spatial and temporal costs of presenting multiple views with spatial and temporal benefits of using the views [Bal00a].

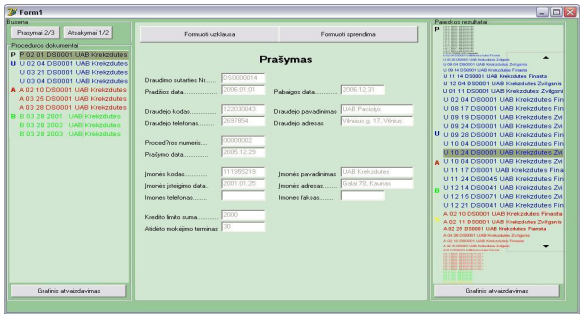


Figure 3. Viewing the application and relevant documents: the combination of multiple-window and fisheye view techniques.

To analyse the time relations between the data timeline style is proposed. The biggest part of the screen displays the set of data using this technique (see Figure 4). This method accords with the rule of complementarity that states that multiple views can be used when different views bring out correlations and disparities [Bal00a].

Complex view can be cognitively overwhelming to the user. The rule of decomposition [Bal00] recommends partitioning to create manageable chunks. In the proposed prototype decomposition is accomplished for the analysis stage. While choosing the request for insurance, on the left pane users can see suspended and new requests. When insurer chooses new request, the header of chosen request is kept to the left pane, the replies to the queries are placed in the right pane. In the middle a detailed view of the chosen reply is shown. When analysis process is completed, the main window with new and suspended requests is shown.

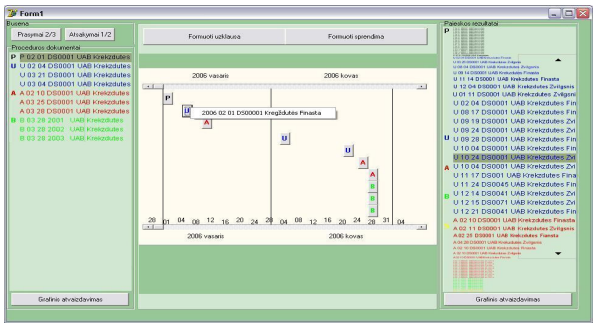


Figure 4. The relevant documents are placed at the timelines: multiply view with timelines

6. EARLY EVALUATION OF THE PROPOSED PROTOTYPE

We claim that combination of multiple views with fisheye and timelines enhances usability of our analysed decision-support system. Usability aspect is manifold. In this paper we measure the advantage of using of the combination of multiple views with timelines and fish-eye. We evaluate the added value using these visualisations in terms of effectiveness and efficiency.

The project is in the initial stage and we propose the prototype of an interface. In order to avoid expensive mistakes, the first evaluation is performed before any implementation. Early usability evaluation often does not involve users directly. Formal expert reviews have proven to be far more effective than informal demos with user testers [Nie94a]. Measuring effectiveness we compare the decision-making process of the current interface with the usage scenario of our prototype. From the variety review methods the cognitive walkthrough and model-based evaluations are based on evaluation of usage procedure. That is the reason why we choose these methods for the early evaluation.

Cognitive walkthrough (CW) was proposed by Polson et el. [Pol92a] and later revised [Wha94a]. This evaluation requires the fairly detailed prototype, description of tasks, list of performed actions and user characteristics. The evaluator steps through the action sequence to critique a system and then tells about the system's usability.

The cognitive walkthrough of the current interface emphasises the five problems. First, it is not visible which field has additional information. Second, the additional information is presented in separate windows and user may feel flustered. It is hard to associate what windows relate with specific request. Third, user experiences the difficulties managing many separate windows. Fourth, when insurer analyses the additional information, the request document is overlapped. It will be accessible, when the additional statistics windows will be closed. Fifth,

 ISBN 978-80-86943-99-2

once began to evaluate the requests insurer cannot analyse the other requests though analysis of specific request is suspended.

The evaluation of the proposed interface endorsed that multiple-window view allowed solving these problems. The whole information space is visible for the chosen request. For each insurance request only related documents are shown and difficulties managing the plenty of windows are avoided. After beginning the analysis in the left pane the header of the chosen document is shown. For suspended request the replied documents that already have been analysed are shown in the left pane. Showing the time relations in timelines facilitate the analysis.

Each document type is associated with specific colours that facilitate distinguishing the documents.

Model-based evaluation

Predictions about user task performance can be made using model-based evaluation, i.e. GOMS model [Car83a]. *Keystroke-level model* (KLM) is lower-level modelling technique that simplifies GOMS assumptions and provides predictions of the time users will take to perform the task with a given prototype [Joh96a]. The advantage of the KLM techniques that it allows a rapid estimate of execution time and can be used *to compare* designs [Car80a].

We compared the existing and proposed designs, using this technique. The evaluation of the existing prototype showed that operations take about 23 seconds to complete the task. In addition to lower-level operations, task performance procedure includes 6 system responses and 6 data analysis steps.

The evaluation of the proposed prototype endorsed that the essential steps take about 16 seconds. Besides these operations the proposed procedure involves 4 system responses and 4 data analysis actions. CW evaluation of data analysis step has shown that problems of existing interfaces are solved.

So we can state that proposed interface is more efficient than the current one because requires less lower-level operations. Evaluation of data analysis step confirmed that proposed interface has solved usability problems caused by existing interface.

7. CONCLUSIONS

Even though the existing decision-support software was created as a means of improving decision-making process and insurance outcomes, the usability evaluation detected many defects. The latter not only frustrate the users, but also influence the effectiveness and efficiency of decision-making process. The combination of multiple views with fisheye and timelines proposed in this paper improves the usability of the analysed system by enabling the user to view the whole information space in different abstract levels and styles. Early evaluations using cognitive walkthrough and keystroke-level model endorsed the benefits of the proposed prototype comparing with the existing system.

8. REFERENCES

[Bal00a] Baldonado, M., Woodruff, A., and Kuchinsky, A. Guidelines for using multiple views in information visualization. Advanced Visual Interfaces (AVI2000), pp.110-119. ACM Press, 2000.

[Bal96a] Ball, T., Eick S.G. Software visualization in the large. Computer 29, No. 4, 1996, pp. 33–43.

[Car80a] Card, S.K., Moran, T.P., and Newell, A. (1980). The keystroke-level model for user performance with interactive systems. Communications of the ACM, 23(7), pp. 396–410.

[Car83a] Card, S.K., Moran, T.P., and Newell, A. The Psychology of Human Computer Interaction, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1983.

[Dix03a] Dix, A., Finlay, J., Abowd, G., and Beale, R. Human-computer interaction: third edition. Prentice Hall, 2003.

[ISO97a] ISO/DIS 9241-11.2 Ergonomics requirements for office work with visualų display terminals (VDTs): Part 11: Guidance on usability, 1997.

[Joh96a] John, B.E., and Kieras, D.E.(1996) The GOMS Family of Analysis Techniques: Comparison and Contrasts. ACM Transactions on Computer-Human Interaction, vol. 3, no. 4, pp. 320–351.

[Nie94a] Nielsen, J., and Mack, R. (Ed.) Usability Inspection Methods. John Wiley & Sons, New York, 1994.

[Pol92a] Polson, P., Lewis, C., Rieman, J., and Wharton, C. Cognitive walkthroughs: A method for theory-based evaluation of user interfaces. International Journal Of Man-Machine Studies, 1992, 36, 741-773.

[Pla96a] Plaisant, C., Milash, B., Rose, A., Widoff, S., Shneiderman, B. LifeLines: Visualizing Personal Histories, Conference on Human Factors and Computing Systems (CHI '96), pp. 221-227, ACM Press, 1996.

[Spe00a] Information Visualization. Pearson Education Higher Education Publishers, 2000.

[Wha94a] Wharton, C., Rieman, J., Lewis, C., and Polson, P. The cognitive walkthrough: a practitioner's guide. Usability Inspection methods. John Wiley, New York, 1994.

POSTER: Topology Maps for Anisotropic Shading

Anders Hast
Creative Media Lab
University of Gävle
Kungsbäcksvägen 47
Sweden, S-801 76, Gävle
aht@hig.se

ABSTRACT

There are a number of materials, like cloth, hair and brushed metal that exhibit anisotropic reflectance properties. Several illumination models have been proposed for shading of such materials. The model by Poulin and Fournier describes how hair, silk and material which can be modeled by parallel cylinders, can be rendered. It has been proposed recently how the topology for this model can vary over the surface as would be the case for many surfaces like spheres and toruses. In this paper it is described how the shading can be visually enhanced by using a map for the topology. The result will be a much more visually pleasing rendering.

Keywords

Anisotropic shading, Varying topology, Mapping.

1. INTRODUCTION

There are a number of materials, like cloth [Tak04, The02], hair [Sta97, Zöc96] and brushed metal [Ash00, Ban94, War92] that exhibit anisotropic reflectance properties. Several illumination models have been proposed for shading of such materials. The model by Poulin and Fournier [Pou90] describes how material which can be modeled by parallel cylinders, can be rendered, like hair and silk. The problem with this model is that the surface is assumed to have equal topology all over the object. In Figure 1 it is obvious that real life objects exhibit a behavior where it is clearly visible that the highlight is affected by the topology of the fibers. This paper will show how such effect can be incorporated in an existing anisotropic shading model.

This is done by using a map for the topology. The result will be a much more visually pleasing rendering since the map makes it possible to vary the height of adjacent cylinders locally as would be the case for a real life object. This approach is used both for the diffuse and the specular light.

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

2. ANISOTROPIC SHADING

Many previous models for anisotropic shading assumes that the topology of the fibers is locally flat. This is shown in figure 2 in the bottom row. It has been proposed recently [Has06] how the topology for this model can vary over the surface as would be the case for many surfaces like spheres and toruses. In the top row in the same figure it is shown how the cylinders start to climb on each other. This behavior occurs when for example a silk thread is winded around a torus or a sphere, like the balls of silk in the photo figure 1.

Figure 1. A real life object where the highlight is clearly non similar because of the topology of the individual fibers.[1]

The area in red is repeated along the adjacent fibers and since this part has equal height locally as its neighbors, it is possible to compute light in this area only. This method uses the shading model proposed

[1] This picture was taken with permission from http://www.thecraftplace.com/

by Poulin and Fournier, however it adds the possibility to vary the topology.

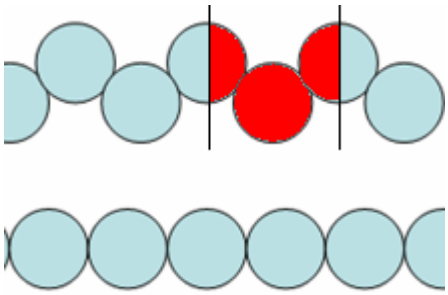


Figure 2. The topology is assumed to be repeated and therefore it is only necessary to compute light in the red area.

There are an infinite number of normals around each cylinder that could be used for shading. It is assumed that there are several fibers visible in each pixel. Hence, the light in a pixel can be approximated by computing the light for one fiber only. Poulin and Fournier proposed that the light contribution for a pixel should be computed by summing the light contribution by using an integral formula that is rather complex. It is possible to solve this integral for the diffuse light analytically. However, for the specular light it is necessary to use an approximation, and they propose one solution using Chebyshev polynomials and another one using a Gaussian distribution.

The diffuse intensity is computed by

$$I_d = \int_{\phi_i}^{\phi_s} (\mathbf{N} \bullet \mathbf{L}) \cos(\phi_E - \phi) d\phi \qquad (1)$$

where the term $\cos(\phi_E - \phi)$ is a correction factor due to the viewing position. Moreover, the intensity must be normalized by dividing by

$$\int_{\phi_i}^{\phi_s} \cos(\phi_E - \phi) d\phi \qquad (2)$$

The more complex model of Poulin and Fournier takes topology of the fibers into account, like distance between fibers and hiding and shadowing, but not the climbing of fibers that is proposed in the varying topology model. If the viewer is looking at the fibers from the side, i.e. not strictly in the tangent direction, then some part of the fiber will be hidden by the fiber closer to the viewer. Similarly, the light will be shadowed by fibers, so that not the whole fiber will be lit. The fiber in the middle of figure 3 is hidden by the fiber to the right. Likewise the fiber in the middle hides the one to the left.

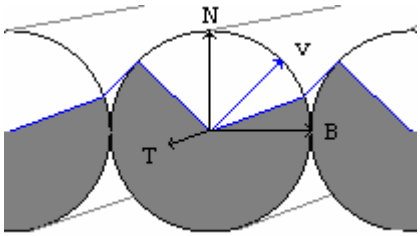


Figure 3. The view vector is pointing to the right. The hiding effect is shown in the figure.

Also note that the view vector does not have to lie in the plane spanned by the normal **N** of the surface and the bi-tangent **B**. Nevertheless it is possible to compute the hiding and shadowing correctly by projecting it down to that plane and perform the computations using the projected vector **v** [Ban94]. Also note that the tangent **T** is perpendicular to both **N** and **B,** and is showing the main direction of the fibers.

3. VARYING TOPOLOGY

When fibers start to climb on each other, both the hiding and shadowing of the fibers will change depending on how high the adjacent fibers are compared to the fiber in question. Since it is assumed that fibers are small compared to the pixel, it is also assumed that all fibers inside a pixel behave in the same way and that the direction to the viewer and light source is constant inside the pixel. The hiding and shadowing of the whole cylinder in the middle will be taken into account when the light contribution is computed. Since the topology is assumed to be repeated, only the right part of the left cylinder is taken into account, as well as the left part of the cylinder to the right as shown in figure 2.

A torus was chosen as test object since it has an outer side that is four times longer than the inner side. Hence, the individual fibers will climb on top of each other two times from the middle of the outer side to the middle of the inner side. The Topology for the torus is shown in figure 4. The height of the cylinders to the left and right of the middle one is computed using the equation

$$H = 2 \sin \theta \qquad (5)$$

where θ varies form 0 º to 60º.

A shading value of zero corresponds to a completely flat topology. As the value increases, the fibers on the sides are being raised (or start to climb). A shading value of 1.0 means the same thing as a shading value of zero, i.e. the fibers on the sides are as close together as possible, i.e. completely adjacent.

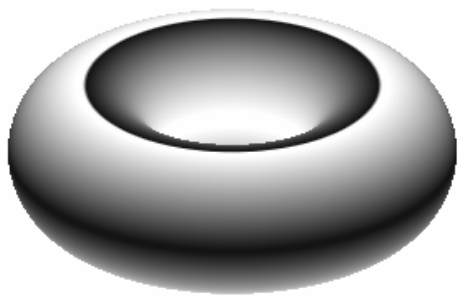


Figure 4. The shading value corresponds to the topology of the fibers on the torus.

4. TOPOLOGY MAPS

The topology can be varied in many ways. In this paper a map was used to make the object appear more real life and not so synthetically smooth. There are many ways to do this. The mathematical topology shown in figure 4 can be changed by adding some noise or adding a regular pattern. Such pattern can be stored in a map and figure 5 shows a map that was chosen for the final renderings. This map is actually a bump map where the height values were scaled down to fit the purpose of being used as a topology map. Actually both the pattern and the mathematical topology can be stored together in one single map, but it was chosen herein to keep them separate. Hence it is possible to distinguish between the two and to choose what kind of perturbation of the mathematical topology we actually want for each case. Different maps were tested, nevertheless, this map gave a quite convincing result.

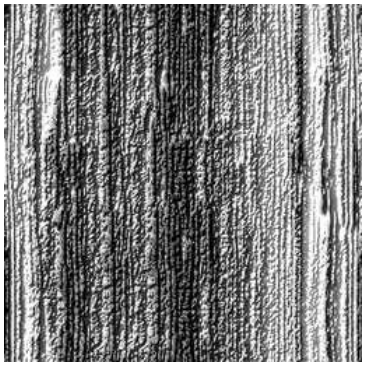


Figure 5. Topology map.[2]

Figure 6 and 7 compares different anisotropic shading techniques under two different lighting conditions. In the top the fast method (Banks) is used where the maximum light is computed. The next row shows the result of the method of Poulin and Fournier where the integral is sampled. The next row shows the varying topology method and finally the last row shows the varying topology method using maps. Note that the varying topology method produces different shadings than the fast method and the Poulin and Fournier method.

The blending of the mathematical topology and the map can be done in many ways. In the renderings the map was scaled so that it was in the range from -0.25 to +0.25. This height was added to the mathematical height and then the result was truncated in the range 0 to $\sqrt{3}$. The total computation was made very simple in this way. For non parametrical objects, i.e. arbitrary polygon objects, it might be necessary to not use a mathematical topology, since it is not always possible to derive it. For these types of objects it is preferable to store the actual topology in the map.

When a map is used it is clear that the shading, especially the highlight, is not so mathematically smooth anymore. It resembles more a real-life object as the ones shown in figure 1. It should be noted that the individual fibers are still assumed to be very small, so that several fibers are visible per pixel.

5. CONCLUSIONS AND FUTURE WORK

This method produces in a very simple way more naturally looking renderings of anisotropic materials. The extra amount of work needed is relatively small compared to the amount of work needed for the varying topology approach in general.

For future work it is proposed that this approach should be investigated further and perhaps a real-time application is possible.

Furthermore, it should be investigated how different level of details can be used so that close ups can be rendered using bump mapping where individual fibers are visible and the map method can be used on larger distances. It should be investigated how a smooth transition between these two LOD's can be performed.

6. REFERENCES

[Ash00] Ashikhmin M, Shirley P, An Anisotropic Phong BRDF Model, Journal of Graphics Tools (2000), 5(2):25-32.

[Ban94] Banks D, Illumination in Diverse Codimensions. In Proceedings SIGGRAPH, 1994 pp. 327–334.

[Has06] A Model for Anisotropic Shading of Fibers with varying Topology A. Hast SCCG'06 full paper, 2006, pp 44-50

[Pou90] Poulin, P, Fournier A, A model for anisotropic reflection. ACM SIGGRAPH Volume 24 Issue 4, 1990 pp. 273–282.

[2] The map used was a free bump map found at http://www.m3corp.com/

[Sta97] Stalling D, Zöckler M, Hege H-C, Fast Display of Illuminated Field Lines, IEEE Transactions on Visualization and Computer Graphics, 1997, Volume 3 , Issue 2 , pp. 118 - 128.

[Tak04] Takeda Y, Quang H, Tanaka H, Image-based Rendering of the Anisotropic BRDF ofWoven Fabrics, EG Workshop on Multimedia (2004) pp.135-143.

[The02] Thevenet L, Dupont D, Caze C. A COMPREHENSIVE PHYSICAL MODEL FOR LIGHT REFLECTION IN TEXTILES FOR COMPUTER GRAPHICS APPLICATIONS, AUTEX Research Journal, Vol. 2, No4, December 2002, pp. 190-198.

[War92] Ward G. Measuring and modeling anisotropic reflection, International Conference on Computer Graphics and Interactive Techniques, Proceedings of the 19th annual conference on Computer graphics and interactive techniques, 1992, pp. 265 – 272.

[Zöc96] Zöckler M, Stalling D and Hege H-C. Interactive visualization of 3D-vector fields using illuminated stream lines. In IEEE Visualization '96, pp 107-113, 1996.

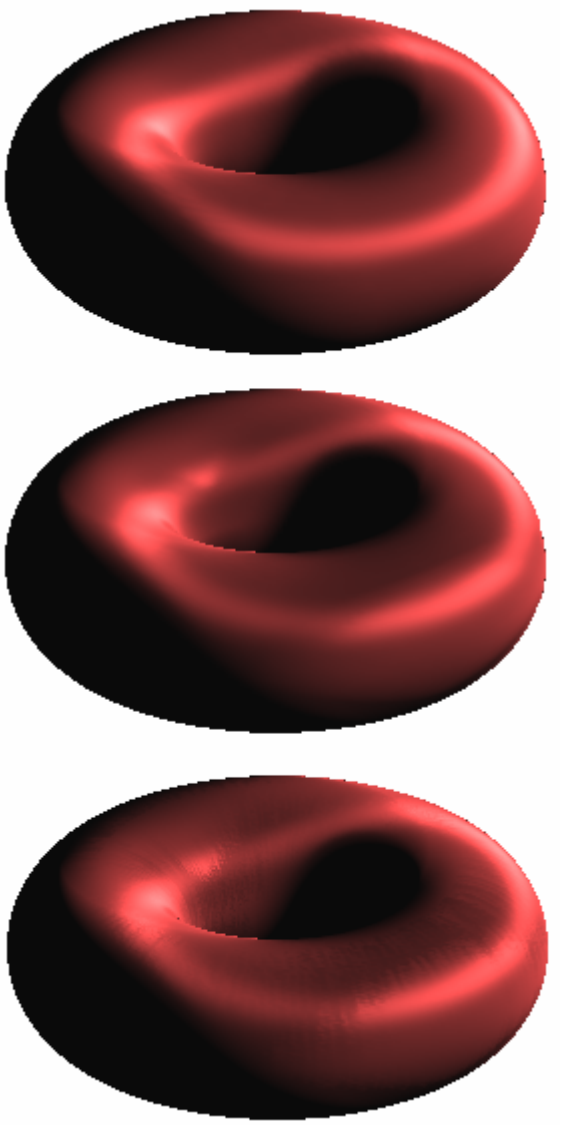


Figure 5. Different anisotropic shading methods. From top to bottom: Poulin and Fournier, Varying topology and Varying topology using a map.

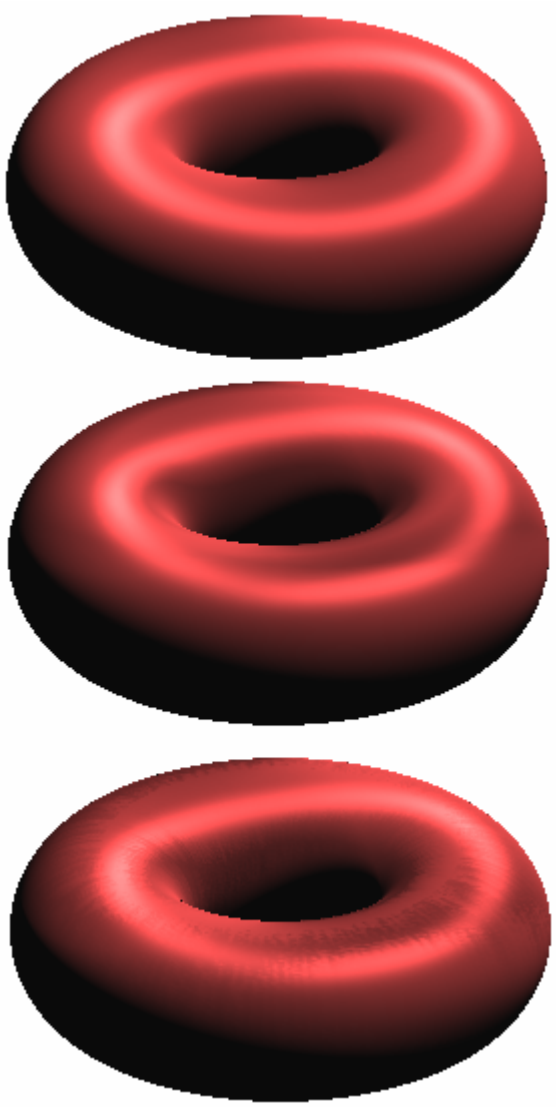


Figure 6. Different anisotropic shading methods. From top to bottom: Poulin and Fournier, Varying topology and Varying topology using a map.

Accelerated Pan and Zoom for Ray Traced Animation

Phil G. Gage and Sudhanhu K. Semwal

Department of Computer Science
University of Colorado, Colorado Springs
CO, 80933-7150
phil.gage@ngc.com | semwal@cs.uccs.edu

ABSTRACT

Ray tracing generates realistic images but is computationally intensive, especially when generating animation sequences. Many techniques have been developed, including exploitation of temporal coherence between successive frames to accelerate animation. This paper presents an efficient pan and zoom algorithm called the Panimation and RSVP algorithm, respectively. This new pan and zoom algorithm integrates the object animation acceleration technique of Jevans with the new pan and zoom algorithms and shows an order of magnitude performance improvement. Short animation sequences are presented to show our results.

Keywords

Ray tracing, animation, rendering.

1. INTRODUCTION

Many acceleration methods have been used to speed up ray tracing of still images, including bounding volumes, spatial subdivision and ray coherence. Jevans [22] presents an elegant method for accelerating ray traced animation using object space temporal coherence (called OSTC in this paper) to track the image areas that need to be updated for each frame. The new pan and zoom algorithms developed in this paper allow camera view changes. The new algorithms can be used separately or combined with other algorithms such as OSTC.

2 PREVIOUS WORK

Image-precision algorithms, such as ray tracing [1,5-8,11-13] have been the topic of much reseach for last several years. Figure 1 provides an example of a ray traced image rendered using our implementation. Spatial subdivision technuiques partition the scene into a grid of volumes called voxels and provide and reduce the rendering time for ray tracing. Some examples are: Glassner [8], Fujimoto [5], Semwal and Dauenhauer [13].

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

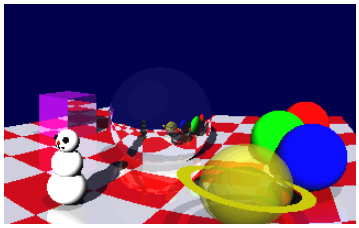


Figure 1. Example Ray Traced Image

Directional techniques [19] such as light buffers, ray coherence, ray classification and proximity clouds make use of direction during the preprocessing. Temporal coherence is discussed in [4] and object space temporal coherence in [9]. Coherence refers to local similarity, or smooth, gradual change in space or time. Chapman [4] presents an image space method similar to 2D morphing for accelerating ray traced animation using temporal coherence. Murakami [11] uses object space coherence with a static view to accelerate ray tracing by redrawing only the changed parts of the image. The ray tree for each pixel is saved, along with additional intersection data.

2.1 Object Space Temporal Coherence (OSTC)

Jevans [8] describes a simpler object space method using spatial subdivision and temporal coherence to accelerate animations with a static camera view. The OSTC method handles object animation well, however, if a light is moved or changes color, only the image areas that are in shadow from both the old and new light location will not be retraced. The shadow rays are kept in a separate bit set in each voxel to improve light animation performance. Briere [3] presents a more advanced approach that allows both color and geometry changes, but requires more complex data structures and algorithms.

3 The PanAnimation (Panimation) algorithm

The obvious way to reuse previous frame data and accelerate pan is to shift the image and redraw only the newly exposed area. Typically, the new area scrolling into view is a small fraction of the image and only a few rows or columns of pixels need to be traced. To allow panning, pixels must cover an equal angle in both directions, implying a spherical projection with equal angle pixels in both azimuth and elevation.

Similar projections are widely used for geographic world maps [12], and several were studied for this application. The Cylindrical Equirectangular projection (Figure 2) was selected as a suitable technique. Using this cylindrical projection or a similar equal-angle method allows pan to be emulated by image shift, since each pixel represents an equal area of the viewing sphere centered at the observer. As the eye recedes an infinite distance from the image plane, both the cylindrical and standard projections approach the same orthographic projection with parallel rays. This indicates that the difference between the cylindrical and standard projections is indeed reduced by narrowing the view angle, which can be confirmed visually.

4 The Ray Sample Viewport (RSVP) algorithm.

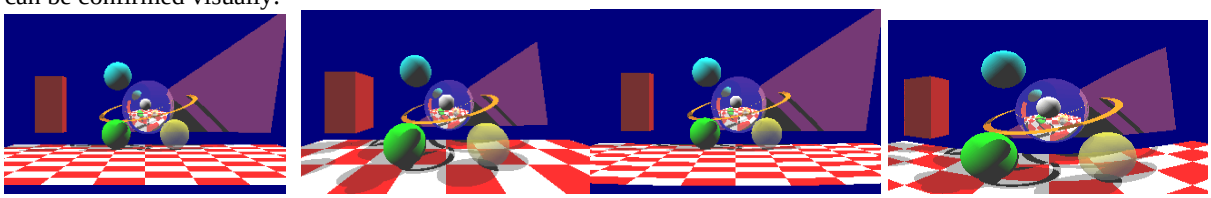
The Panimation algorithm does not support accelerated zoom, so we wanted to develop a better zoom algorithm to allow smooth zoom at any factor. A better approach for zooming out is to draw the new pixels around the edges of the image at the same resolution as the original image and resample the whole area down to a lower resolution for each frame. We call this approachh the Ray Sample Viewport (RSVP) algorithm.

Other useful effects are supported by the RSVP method. Rotate, scale, shear and more unusual changes can be produced by simple 2D transforms. Any pattern of pixels in the camera frame buffer can be resampled into the viewport image. This can accelerate pan, zoom and other transforms within the frame buffer view area.

5 Implementation

We implemented three algorithms in this paper, the Jevans OSTC animation acceleration method [9], the new Panimation algorithm and the new RSVP algorithm.

Integrating the OSTC and RSVP algorithms is simple. Instead of ray tracing the OSTC updated image areas, they are erased to the null color. When a viewport samples a null pixel, it is ray traced at that time. Since the tan function is used for the cylindrical projection, the arctan (or atan2) function can be used for the inverse transform, restoring the standard projection appearance. This fixes a major problem with the Panimation algorithm. Panimation provides fast large-scale pan but causes unwanted curvilinear distortion. Adding RSVP provides fast small-scale pan, zoom and other effects. The straight correction by our algorithm of the l checkerboard is shown in Figure 3. Java [2] was selected because of its OO features, graphic libraries and portability.



(a) Standard Projection 30° Angle b) Standard Projection 100° Angle c) Cylindrical Projection 30° Angle d) Cylindrical Projection 100° Angle

Figure 2. Standard and Cylindrical Projections

 ISBN 978-80-86943-99-2

6 Results and Discussion

The Test program generates a sequence of animation frame TGA files and provides Boolean flags to control animation and acceleration features. The OSTC method uses 16x16 screen regions with 256 bits in each bitset, the same as Jevans [9]. The scene contains a checkerboard ground plane, 12 polygons, 19 spheres and 2 light sources. A cylindrical camera is used with field of view 100 degrees.

Our results show that panning 3.6 degrees per frame was accelerated using Panimation by about a factor of 10. Zooming by a 2:1 ratio was accelerated using RSVP by about a factor of 10. Animation of a small bouncing ball in this scene was accelerated using OSTC by about a factor of 2. Simultaneous pan, zoom and animation caused complex view changes and was accelerated by about a factor of 2. These techniques can be used to accelerate pan, zoom and animation simultaneously. These results are very encouraging and indicate that even greater performance can be achieved for slower pan and zoom or longer animation sequences. The resulting 100 frame animation sequences were converted to animated GIF files and are available for review.

Figure 4 shows a test image where the black areas are rectangles erased to the null color by pan and object animation. The narrow black strips at the top and right are caused by the camera panning slightly up and right, shifting the image down and left. Only the black areas need to be ray traced, the majority of the image is reused from the previous frame, showing the potential for image space acceleration even with simultaneous pan and object animation. Figure 5 shows a camera image and two smaller viewport images extracted from the camera image. In the camera image, only the pixels sampled by the viewports were ray traced, the black areas are null pixels not traced.

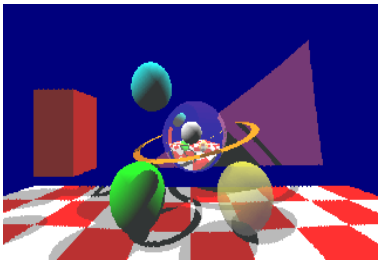


Figure 3: Corrected Cylindrical distortion (above)

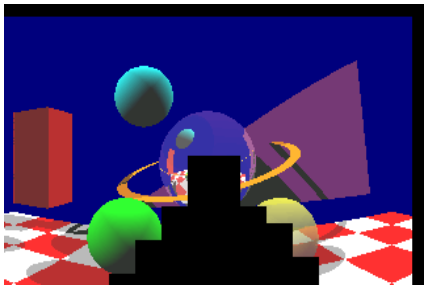
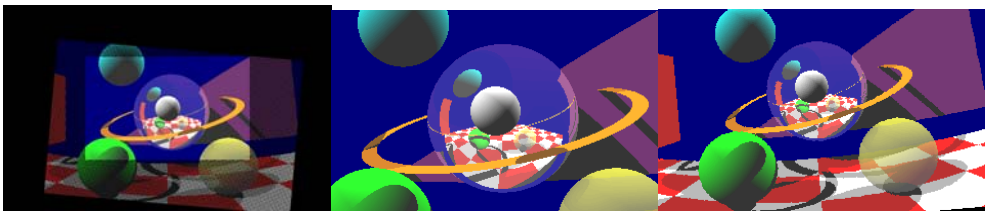


Figure 4. Simultaneous Pan and Animation Image (right)



a.) 640x480 camera image with sampled pixels. b.) 320x200 sampled zoom viewport c.) 320x200 zoom and rotate viewport

Figure 5. Camera and Viewport Images

 ISBN 978-80-86943-99-2

7 FUTURE WORK

This paper has focused on acceleration, but many other features could be added. Realism could be improved by interpolating sampled pixel colors, antialiasing, distribution ray tracing and improved texture mapping. It would be good to trigger auto-pan and auto-zoom events when a viewport hits a buffer edge or size limit. Other view projections such as Fisheye projections [14] would be interesting to implement.

8 CONCLUSION

This paper has presented the history and theory of ray tracing and animation acceleration. The static camera limitation of Jevans [9] and other earlier work was noted and animage space view acceleration algorithm was developed to accelerate pan, zoom and other effects from a fixed viewpoint. Fast pan is performed using image shifting with an equal-angle view projection. Fast zoom and other effects are performed using a sampling viewport and 2D transforms. Implementation details, more comparisons and rendering results of our implementation are in [15].

A Java [2] ray tracer was implemented to test the new algorithms both for animation and interaction. An object space temporal coherence animation acceleration method based on the Jevans [9] technique was also implemented and integrated with the new algorithms. The resulting system supports ray traced animation with acceleration for both object changes and camera field of view changes. A generalized animation methodology was developed that could be extended to include reprojection, saved ray trees and other accelerations.

The timing results show performance gains of an order of magnitude for the new pan and zoom algorithms. This work has been successful, personally rewarding, and has many applications and possibilities for future work. Possible uses include animation, interactive scene editing, virtual reality, video games, simulations, computer-aided design and 3D data visualization.

9 REFERENCES

[1] Amantides, J., Woo, A., "A Fast Voxel Traversal Algorithm for Ray Tracing," Proceedings of Eurographics, 1987, pp. 3-10

[2] Arnold, K., Gosling, J., The Java Programming Language, Addison-Wesley, 1998

[3] Briere, N., Poulin, P., "Hierarchical View-dependent Structures for Interactive Scene Manipulation," Computer Graphics Proceedings, 1996, pp. 83-90

[4] Chapman, J., Calvert, T., "Exploiting Temporal Coherence in Ray Tracing," Proceedings of Graphics Interface 90, 1990, pp. 196-204

[5] Fujimoto, A., Tanaka, T., Iwata, K., "ARTS: Accelerated Ray Tracing System," IEEE Computer Graphics & Applications, Apr 1986, pp. 16-26

[6] Gassner, A., "Space Subdivision for Fast Ray Tracing," IEEE Computer Graphics & Applications, Oct 1984, Vol. 4, No. 10, pp. 15-22

[7] Glassner, A., An Introduction to Ray Tracing, Academic Press, 1989

[8] Jevans, D., "Adaptive Voxel Subdivision for Ray Tracing," Proceedings of Graphics Interface 89, 1989, pp. 164-172

[9] Jevans, D., "Object Space Temporal Coherence for Ray Tracing," Proceedings of Graphics Interface 92, May 1992, pp. 176-183

[10] Lext, J., Assarsson, U., Moller, T., "BART: A Benchmark for Animated Ray Tracing," IEEE Computer Graphics & Applications, March 2001, Vol. 21, No. 2, pp. 22-31

[11] Murakami, K., Hirota, K., "Incremental Ray Tracing," Proceedings of Eurographics Workshop on Photosimulation, June 1990, pp. 15-29

[12] Musgrave, K., "A Panoramic Virtual Screen for Ray Tracing," Graphic Gems III, Academic Press, 1992, pp. 288-294

[13] Semwal, S. and Dauenhauer D., "Approximate Ray Tracing," Proceedings of the Graphics Interface, Halifax, Nova Scotia, pp. 75-82 (1990).

[14] Glaeser, G., Groller, E., "Fast Generation of Curved Perspectives for Ultra-Wide-Angle Lenses in VR Applications," The Visual Computer, July 1999

[15] Phil Gage "View Coherecnce Acceleration for Ray Traced Animation," MS Thesis, Univesity of Colorado, Colorado Springs, CO, US, directed by Professor Semwal, pp. 1-75. 2002.

POSTER: Duct tracking in 3D medical data

Martin Petříček
Faculty of Mathematics and Physics
Charles University, Prague
Malostranské nám. 25, 11800 Praha 1,
Czech Republic
martin@petricek.net

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

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

ABSTRACT

Implementing duct tracking of bronchi and blood vessels in 3D medical data can significantly help in examining these body structures and reduce time needed for correct diagnosis. Nature of the problem involved may require some operator interaction when selecting duct branches of interest or correcting possible mistakes in duct path segmentation. Noise, badly discernible duct walls, branching or extreme duct curvatures, that can be found in computer tomography data, make the segmentation challenging. There are more ways to display segmented ducts, as overlay over original data or unrolled to planar tube, allowing easier measurements of length, width or observation of tube anomalies.

Keywords: Image segmentation, visualization, 3D medical data, computer tomography.

1 INTRODUCTION

Examination of human body with computer tomography or magnetic resonance allows to see inside a patient. Unlike X-ray imaging these measurements provide full 3D data, which requires specialized program to view or visualize them.

In this paper we specialize on duct tracking. This include ducts in respiratory system (trachea, bronchi), blood vessels (veins, arteries) and possibly gastrointestinal tract.

First type are bronchi - tubular structures in respiratory system, usually filled with air, although they can be relatively commonly blocked by various inhaled objects. Bronchi are probably easier for segmentation, as they usually contain only air and are the simplest to segment.

Second type of ducts in human body are blood vessels - veins and arteries. They are used to transport blood in human body. Sometimes a contrast agent, such as Ultravist, is applied to enhance the density of blood in veins. This make them more contrast and thus more visible, as we can see in example in Figure 2. In the image without contrast agent the artery blends with surrounding tissues of similar density. With contrast agent applied, the artery is clearly visible. Quite common feature that can be found in blood vessels is thrombus. The thrombus has almost the same density as blood, so it is generally visible only with help of contrast agent.

Figure 1: Following ducts in CT data can be tedious.

Third type is gastrointestinal tract (small and large intestine). These are the hardest to segment. They can contain liquid or solid content, or even pockets of gas, which can make segmentation problematic, as they have highly varying densities. Intestines have thin, badly visible walls, the curvature and circumference is varying largely as the intestine continues through the body and the intestines often touch themselves as they are folded in the abdominal area. Example of slice showing part of gastrointestinal tract can be seen in Figure 3

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

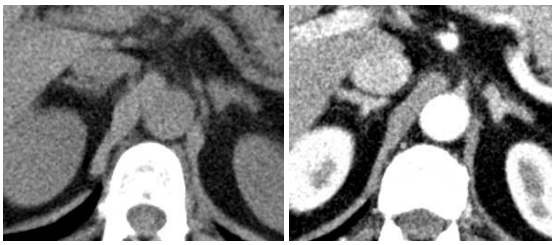


Figure 2: Blood vessel (aorta) - left image is slice from data without contrast agent, right image is from data with contrast agent (Ultravist) applied

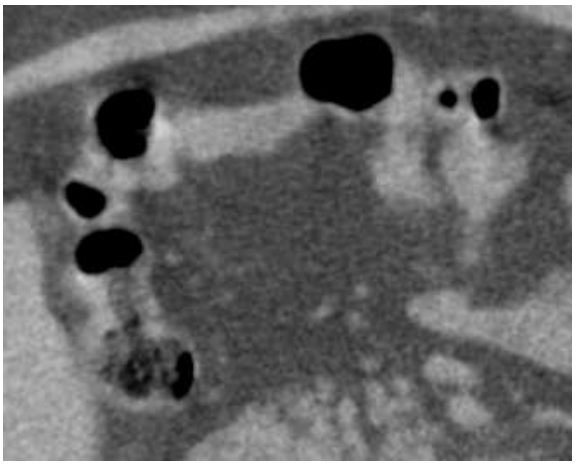


Figure 3: Image from gastrointestinal tract with visible pockets of gas, example of very hard segmentation problem

2 PROBLEM DECOMPOSITION

We have analyzed the problem and decomposed it into multiple consecutive steps.

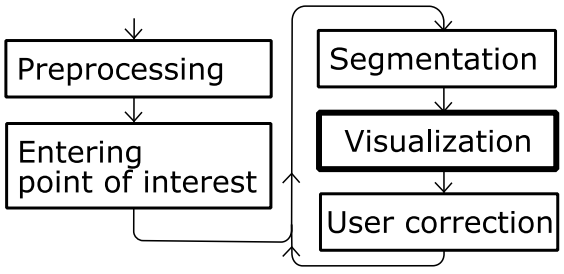


Figure 4: Problem decomposition into separate steps

First optional step is preprocessing. It may improve the segmentation and the resulting image if some filters to reduce noise are used, as CT and MRI tend to be generally noisy. Three dimensional variant of Averaging using a rotating mask [SHB93] with $3 \times 3 \times 3$ masks in $5 \times 5 \times 5$ area can be used, as it preserve edges. Its disadvantage is computational intensity, because for each voxel, brightness dispersion σ^2 needs to be computed for each rotated mask. Median filtering is faster, for $3 \times 3 \times 3$ mask by factor of 7.30, for $5 \times 5 \times 5$ by factor of 1.65. Disadvantage of median filtering can be rounding of sharp corners, but that might not be noticeable in most anatomy images. Both noise-removing filters can be run on series of slices as 2D filters, or extended to fully 3D filter.

As the image is usually relatively much larger than the area of interest, it is possible to reduce time needed for preprocessing. Preprocessing can be run only on a small area around the starting point and then on neighbor areas, as the segmentation continues and tracks the duct through the body. As result, the computationally expensive filter will be run only on areas that are close to the area of interest.

Second step is entering point of interest. Basically, there are many ducts in human body and user must pick some starting point inside the duct in which he is interested and direction in which to track it.

For some types of images and few specific cases, the starting point can be recognized automatically and may be offered as a preselection (like position of trachea on image from chest area).

Third step is the segmentation. The hard part is finding the duct walls. Once duct walls are identified, finding the axis (centerline) and inside volume is quite straightforward.

Segmentation and visualization will be discussed with greater detail in next two chapters

Fourth step is visualization. There are more possible ways for showing resulting ducts. Either they can be highlighted in 3D view of original data or they can be unwinded and displayed separately. Unwinded duct still remain a 3D image, though.

Fifth and the last step is correction. As ducts in body often have branches, user have to pick one of the ways. Also, there will be cases in which the segmentation was incorrect and algorithm has left the duct. These should be corrected. The algorithm should continue following the proper duct after adjustment, basically running again on part of the duct from third step with modified parameters. This requires functions in visualization part that enable user to specify the corrections.

Another possibility is to segment entire duct tree, forking the segmentation at every duct split. Entire bronchi system can be segmented and then displayed or measured this way.

3 SEGMENTATION

Segmentation is still work in progress. Most promising are Active Shape models, also called "snakes": [CT92], [dBvGVN03], or its variant, Active Appearance Models [CET98]. There are many variations to these methods: [XP97], [CC93], [OBFG94], [Coh91].

They are contour models, attracted to image features, such as edges. The contour is usually represented as a spline curve and often there are imposed limits on the deformation to ensure it maintain "reasonable" shape.

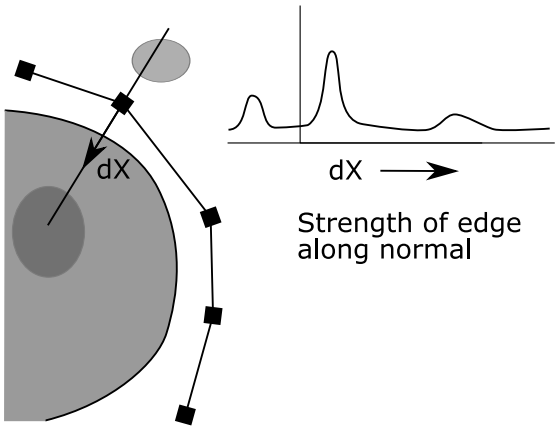


Figure 5: Points on the curve are moved towards strongest edge, proportional to its strength. The process is repeated for each point until the curve converges to approximate the object contour

In the algorithm, first the initial curve is placed in the data - for tubes it will be initially a circle, or shape of the tube from previous slice. Then, the algorithm step is repeated, until the model converges. In each step, every point is moved towards strongest nearby edge along the normal of the contour in that point. This results in the contour following the edge.

The model needs to be extended to 3D data and slices that are generally not parallel to any major axis or to other slices.

Another possible approach is not to use 2D deformable models on moving slices, but use instead a 3D deformable model (possibly trying to fit a cylinder on the duct walls). Advantage of such approach is having surface representation directly as the result and we don't have to deal with artifacts that can result from extracting slices of arbitrary orientation from the data. As edge detector is part of the method, we also have to deal with data anisotropy. Very often the resolution of the data is not uniform, the voxel size in z axis, corresponding to the gap between image slices, is larger than in x and y axes. This is usually is problematic for many edge detectors. We may solve it by resampling the data, modifying the edge detector, or we can use edge detector that can work with anisotropic data, like method proposed by Brejl and Sonka[BS00].

More methods [Lak00] will be examined before deciding which one will be finally used.

For finding the first part of the duct, some other way must be used, as we have no previous duct slice. The position and direction may be somewhat inaccurate, as the user probably won't pinpoint exact center of the duct and its precise direction. We can use a variant of the method, where we allow tilting of the initial plane and try to minimize length of the curve (assuming minimal length is at the tube cross-section, this will compensate for inaccurate initial direction). After successful segmentation, we can update the position to correspond to the centerline of the duct found. Sometimes it is hard to tell exactly what center is, but we use center of mass calculated from the duct walls for this purpose.

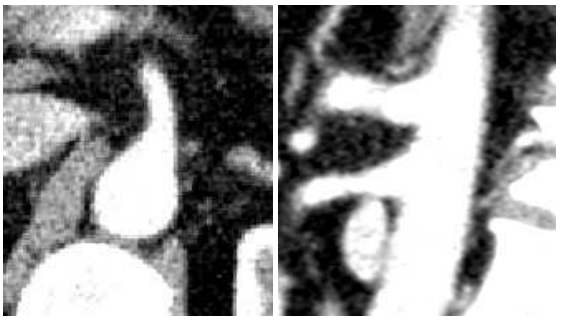


Figure 6: Branching of aorta, with contrast agent applied. Left image show situation from above, right image shows it from side

$p_0 = initial\ position$
$d_0 = initial\ direction$
$d_0 = correct\ orientation(p_0, d_0)$
$w_0 = find\ duct\ walls(p_0, d_0)$
$p_0 = duct\ center(w_0)$
for $i = 1..length\ of\ duct$ **do**
  $p_i = p_{i-1} + d_{i-1}$
  $d_i = d_{i-1}$
  $w_i = find\ duct\ walls(p_i, d_i, w_{i-1})$
  $p_i = duct\ center(w_i)$
  $d_i = p_i - p_{i-1}$
$Array\ p_i\ holds\ positions$
$Array\ d_i\ hold\ directions$
$Array\ w_i\ holds\ walls\ slices$

Algorithm 1: Pseudo-code of segmentation algorithm. "find duct walls" is the main algorithm to segment the walls, given the 2D slice specified by (p,d) and position of duct walls in previous slice (except the first slice, where the initial duct must be found some other way)

Result of duct segmentation from Algorithm 1 will be the centerline represented by series of points located approximately 1 voxel apart. A duct direction vector is defined in every such point. It is used as normal vector of plane for extracting a slice from the data. On each of these planes a border will be segmented, using information from last plane, since the planes are close to each other. As a result of one plane segmentation, centerline position and direction will be updated for next step.

By approximating the curves from all slices as a polygon and joining them it is possible to build polygonal 3D model of the duct. Alternatively, we can use B-splines, NURBS or similar smooth surface representation to get nicer results.

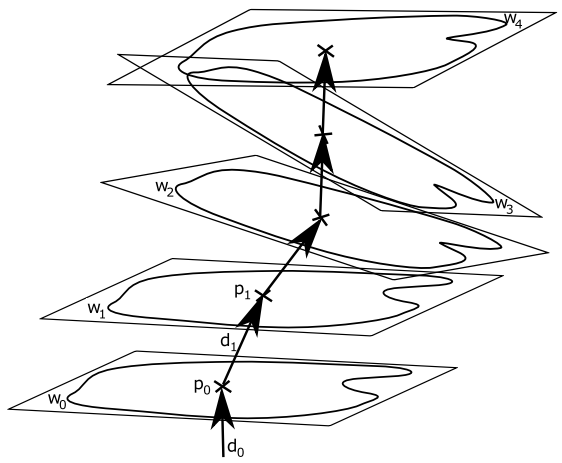


Figure 7: Example of one segmentation algorithm run, as specified by Algorithm 1. The segmentation starts on bottommost slice and moves upwards.

4 VISUALIZATION

There are two visualization modes, 3D mode and "duct unwinding" mode. In 3D mode, the segmented ducts will be highlighted in some way. Either by coloring the ducts, or by modifying transparency - so the ducts will be almost opaque, while the rest will be almost completely transparent. Both approaches could be combined. For the 3D mode, we will use part of MedV3D framework designed to display volume data.

In the second mode, slices are created along the duct center. Slices can be relatively small (width of the duct and some small part of surrounding tissue), but number of slices can be relatively large, if the duct is long. They are still displayed as 3D data, but in more comprehensible form.

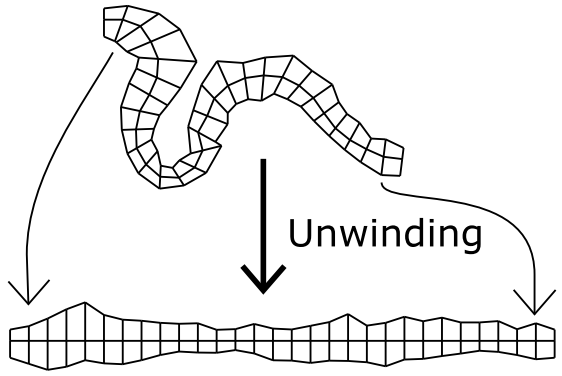


Figure 8: Schematic of "Duct unwinding" mode: Duct in original position is shown above, while corresponding unwinded duct is shown below. Length of centerline remain the same.

The deformation caused by unwinding will preserve lengths in each of the slices created and also length of the duct center will be preserved. Other lengths may be more or less deformed by the transformation, greatest deformation is in points farthest from centerline and in duct segments with greatest curvature.

Also, in this mode, specifying the corrections should be much easier than in ordinary 3D mode, as in case of bad segmentation, the correct duct leaving the resulting image can be easily spotted and corrected. See Figure 8 for simplified 2D schematic of how this mode will work. Once the duct is segmented, the data can be measured (for example volume of duct between certain points, its circumference, length, etc ...), as the voxel size is calibrated and its dimensions are part of the input data.

5 CONCLUSION AND FUTURE WORK

We have presented method for segmenting, measuring and displaying ducts in medical 3D images using deformable models, dealing with problems like anisotropy of the data, noise and irregularities in bodily ducts. The described approach aim to reduce time needed for proper patient diagnosis if bodily ducts are involved.

ACKNOWLEDGMENTS

This work is part of MedV3D framework, supported by GAUK grant No. 359/2006 (Charles University in Prague). Thanks to Martin Horák for helping with medical data.

REFERENCES

[BS00] Brejl, M. and Sonka, M. (2000). Directional 3d edge detection in anisotropic data: detector design and performance assessment. *Comput. Vis. Image Underst.*, 77(9):84–110.

[CC93] Cohen, L. and Cohen, I. (1993). Finite-element methods for active contour models and balloons for 2-d and 3-d images. *PAMI*, 15(11):1131–1147.

[Coh91] Cohen, L. D. (1991). On active contour models and balloons. *Computer Vision, Graphics, and Image Processing. Image Understanding*, 53(2):211–218.

[CT92] Cootes, T. and Taylor, C. (1992). Active shape models- - smart snakes.

[CET98] Cootes, T. F., Edwards, G. J., and Taylor, C. J. (1998). Active appearance models. *Lecture Notes in Computer Science*, 1407:484–??

[dBvGVN03] de Bruijne, M., van Ginneken, B., Viergever, M., and Niessen, W. (2003). Adapting active shape models for 3d segmentation of tubular structures in medical images.

[Lak00] Lakare, S. (2000). 3d segmentation techniques for medical volumes.

[OBFG94] O'Donnell, T., Boult, T., Fang, X., and Gupta, A. (1994). The extruded generalized cylinder: A deformable model for object recovery.

[SHB93] Sonka, M., Hlavac, V., and Boyle, R. (1993). *Image Processing, Analysis, and Machine Vision*. Chapman and Hall, London, New York.

[XP97] Xu, C. and Prince, J. (1997). Gradient vector flow: A new external force for snakes. In *Proceedings of Computer Vision and Pattern Recognition (CVPR '97)*, pages 66–71, San Juan, Puerto Rico. IEEE.

A Curvilinear Cutting Tool for the Analysis of Oceanography Data

Nathalie Rey da Silva
PUCRS - Pontifícia Universidade Católica do Rio Grande do Sul, Faculdade de Informática
Av. Ipiranga, 6681 Prédio 30, Bloco C
90619-900, Porto Alegre/RS, Brazil
nrsilva@inf.pucrs.br

Isabel Harb Manssour
PUCRS - Pontifícia Universidade Católica do Rio Grande do Sul, Faculdade de Informática
Av. Ipiranga, 6681 Prédio 30, Bloco C
90619-900, Porto Alegre/RS, Brazil
manssour@inf.pucrs.br

ABSTRACT

The use of new technologies for the ocean data acquisition produced a considerable increase of information. Thus, scientific visualization tools become essential to aid the oceanographers during the analysis and interpretation of ocean behavior. The main goal of this paper is to present the design and development of a set of tools for 3D visualization of simulated oceanic data, including a new approach for the selection of regions of interest using a curvilinear clipping tool. The implementation is based on a study about the tasks developed by the oceanographers of the Research Center of Petrobras.

Keywords

Visualization, Oceanography, Curvilinear Clipping Tool, Region of Interest.

1. INTRODUCTION

The ocean is a 3D space that changes as the time passes and is inhabited by living beings of all scales (from micro-organisms to whales), being subject to a great variety of force functions, including seismic, thermodynamic, atmospheric, inertial and maritime [Ros89a]. The study of the ocean is important for the climatic and meteorological aspects. There are also great solid mineral deposits, as diamonds, and soluble minerals, as oil and gas, deep in the ocean.

In the Physical Oceanography, many oceanographic parameters are involved, such as salinity, density, temperature, speeds of chains, pressure and pH. These data can be obtained through: simulation, where numerical models are processed to generate information about the evolution of the oceanic features as the time passes; or using data collection, where buoys and drills, and even satellites, catch real data from the ocean in definite intervals of time.

Tools and techniques of scientific visualization are being used for the analysis of the oceanic data. Through the visualization, the scientists can observe the changes of the ocean and aid in the forecasts about its behavior. Visualization techniques are used for physical oceanic data taking into account the kind of data, since it can be scalar, vector or even tensor. Polito [Pol04a] says that we seldom need to describe oceanographic variables as tensors. The geometric disposal and the space-weather scale of the geophysical flow also simplify the treatment of tensors variable. Directions,surfaces and symmetries, due to the macroscopic treatment, reduce the number of interesting components of the tensors. Thus, the majority of the cases deal with vectors and scalars.

With the intention of developing a scientific visualization tool, the Research Center of Petrobras (CENPES) was contacted. The oceanographers of CENPES had shown interest in contributing to the visualization techniques research for oceanographic data generated by simulation. The visualization of simulated data could help these professionals in the understanding of the ocean circulation, such as running and temperature, and of the licensing and ambient impact in the oil extraction. Moreover, these ambient data can be applied for the analysis of the oil spilling and sediments transport.

The main goal of this paper is to present the design and development of a prototype that provides a set of tools for 3D visualization of simulated oceanic data, including a new approach for the selection of regions-of-interest (ROI) using a curvilinear clipping tool. All the implementation is based on a study about the tasks developed by the oceanographers of CENPES. As a case study to experiment the implemented tools, we have used a simulated data obtained at CENPES.

The paper is organized as follows: Section 2 introduce some visualization techniques. The

developed prototype is presented in Section 3. Section 4 describes the curvilinear clipping tool. Conclusions and future work are presented in Section 5.

2. VISUALIZATION TECHNIQUES

There are several 2D and 3D visualization techniques for scalar and vector ocean data. According to Head [Hea97a] 2D techniques are enough for a lot of tasks, revision and planning in data analysis. However, a better understanding of the dataset and a reduction of the analysis time can be obtained including 3D visualization techniques.

A number of powerful techniques were already developed for the visualization of vector data. Techniques like Streamlines and Stream Tubes show the way that a particle follows in a fluid. Methods based on icons, like Hedgehogs [Gai02a], show the direction and the magnitude of a vector field in selected points. Methods as spot noise and line integral convolution (LIC) use texture to show the fluid direction on surfaces [Cro01a]. For scalar data, techniques as color mapping [Ebe06a] and isosurfaces are commonly used.

3. PROTOTYPE DESCRIPTION

Firstly we got in touch with the oceanographers of CENPES. Based in their work and necessities, we have made a research and developed a tool for the analysis of the oceanography data.

The prototype developed in this work allows the 3D visualization of the ocean, with the possibility to load data files in the NetCDF format [Rew90a] generated by the Princeton Ocean Model (POM). The data, as well as system functionalities, are described in the next Sections.

Oceanography Data

The simulation generates a great quantity of information about the ocean, allowing to identify and diagnose problems and to learn more about its behavior. Data models implemented in some programming languages as Fortran or C, are used to do the simulation. After processing, the results are sent to output files, using an appropriate data format, for analysis and visualization. Several data formats are used to store the ocean data. The oceanography applications usually work with HDF [Nat06a], NetCDF [Rew90a], CDF [Cdf05a], among others.

In this work we are using the data generated by the POM model, in the NetCDF format. These data, which had been supplied by CENPES, correspond to a ten days simulation done through Bacia de Campos, in Rio de Janeiro. Its size is 609 MB taking into account a volumetric grid. For 3D grid generation 630000 cells are used.

POM is a sigma coordinated model where the vertical coordinate represents the depth of the water column. This coordinate system is an important attribute to deal with the existent topographical variability [Mel03a]. The coordinate sigma is defined in equation 1, where z represents the Cartesian coordinate and d represents the depth of the water column, with σ ranging from 0 to -1 (with 21 sigma levels). As a result, the water column always has 21 levels, as illustrated in Figure 1, where η is the surface elevation, h is the height of the water column and σ corresponds to the sigma level.

$$\sigma = z / d \qquad (1)$$

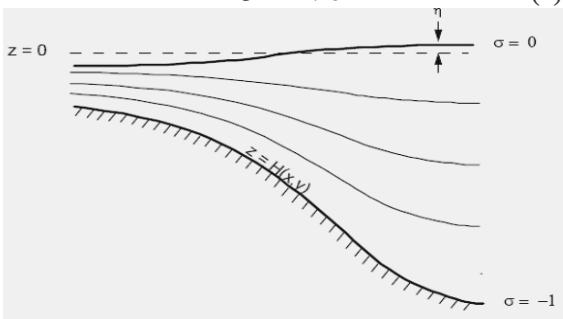


Figure 1. Sigma Coordinate system [Mel03a].

Implementation and Functionalities

Our prototype was developed for Windows platform using C++ programming language and the Visualization ToolKit (VTK) version 5.0. A library developed by Unidata [Uni05a] was used to read the NetCDF data. The graphical user interface was implemented with the Microsoft Foundation Classes.

Among the implemented functionalities, the application provides vector and scalar visualization techniques. Figure 2 shows an example of the temperature variable visualization using RGB color mapping. In this volume the temperature values are mapped in five ranges between 0 and 30 Celsius degrees, with a color corresponding to each temperature range.

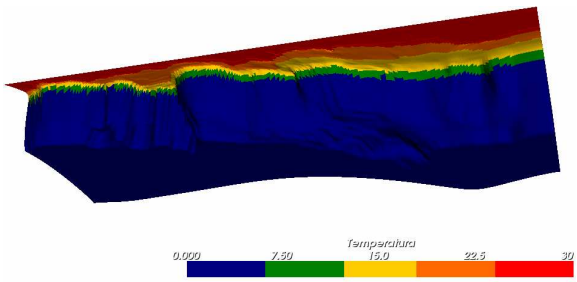


Figure 2. Temperature visualization.

 ISBN 978-80-86943-99-2

For vector visualization we applied the *Hedgehogs* technique. In this case, the 3D arrows point to vector direction and its magnitude is represented through arrow size and color mapping. Figure 3 shows an example of the image generated by our prototype using the *Hedgehogs* technique. In this case, the speed magnitude is given by the length of the arrow and mapping of colors. The arrow is guided in agreement with the direction of the chains speed.

Interaction methods present in the application make possible to obtain a better understanding of volumetric data. For example, it is possible to translate, rotate and to apply zoom in and zoom out in the volume. The user can also select a ROI through splines manipulation [Bar87a] (Section 4).

Besides the volume visualization, our prototype also provides an animation tool. The simulated data are obtained in time steps that correspond to days of simulation. To analyze this temporary variation, we developed a technique to update the position and the state of each variable along the time. This can be done by reading the oceanic variable in all the time steps and applying the appropriate visualization technique for it. Thus, we provide a way to observe the changes in the variable behavior.

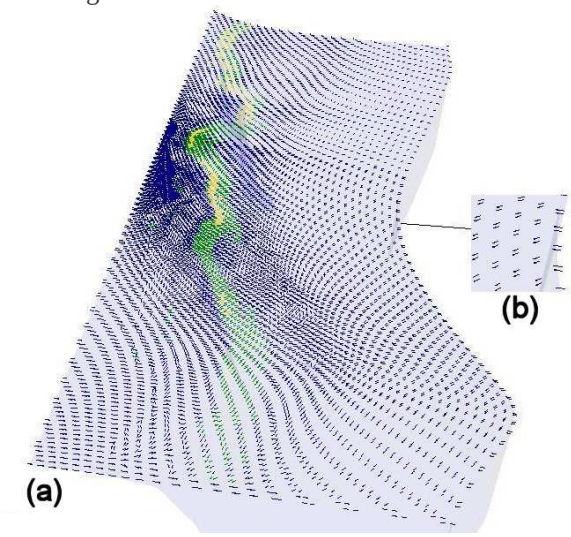


Figure 3. Example of vector visualization for the chains speed of the total volume (b) and applying zoom (a).

4. CURVILINEAR CLIPPING TOOL

The developed clipping technique allows to select a 3D ROI and to interact with the volume providing a better way to analyze the ocean in a 3D visualization. This clipping tool, as well as the other visualization tools, doesn't depend on the NetCDF format. Since a 3D grid is created, any other format could be used if an appropriate reader is used to generate the 3D grid, where the clipping is done.

The Curvilinear Clipping tool allows clipping the volume using splines. In order to facilitate ROI selection, several control points can be manipulated to specify the curve that will be used to clip. In the example presented in Figure 4, the spline is manipulated in a surface that corresponds to the *xy* plane. In this case, several control points of the segments of lines that compose the spline are manipulated by the user. After, the Kochanek interpolation [Koc84a] is used to specify the spline curve, allowing a better control of its shape. Figure 4a shows a spline with 9 control points positioned over the visualized volume where the colors represent the temperature.

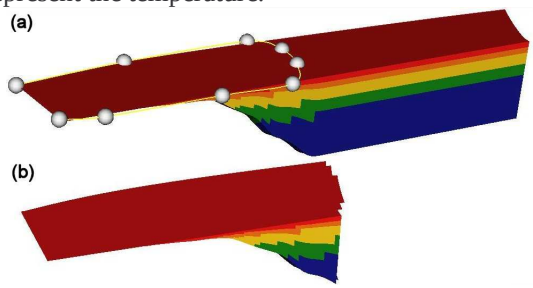


Figure 4. (a) Spline position in the volume after its manipulation; (b) ROI selected with the spline specified in (a).

After spline specification the sub-volume is generated, as shown in Figure 4b. The ROI is defined dragging the spline along the *z* axis, as if it was a knife. The implemented clipping algorithm selects the cells of the volume that collide with the spline and also the cells that are inside the selection area (along the *z* axis in the presented example).

The identification of the cells that collide with the defined points of the splines is a little complex. One of the alternatives that was implemented, is to seek in all cells and verify which one contains the searched for point *p*. For this, cell interpolation functions are resolved for the parametric coordinates. If these coordinates are inside the cell, then *p* is in the cell. To guarantee a better performance, data structures that support spatial search must be used (e.g. octrees and Kd-trees).

Thus, using the spatial search in octrees available in VTK, and then obtaining all the cells that collide with the splines, it is necessary to acquire the other cells that are part of the selection. The grid cells identifiers are increasing ordered from left to right, from bottom to top, and are attributed during the grid structure creation as natural numbers(Figure 5).

To do the clipping, the ends of the spline should be positioned in the left or right border of the surface, as indicated in Figure 4a. Initially, the identifiers of the surface cells that collide with the spline are

obtained, storing them in a list. This algorithm is executed until all cell identifiers of this list are read.

The clipping algorithm works in the following way:

1. Read a value of the cells list.
2. Taking into account the cell identifier from the list, firstly test if the cell right bellow (column) is part of the border and if it is smaller than the largest identifier value of the list. In a positive case, this identifier is part of the clipping. A new identifier is read from list (step 1). Otherwise, follows the step 3.
3. Starting from the cell identifier of the list, test if the cell right bellow (column) is part of the list, or if the neighbor of this cell right below (left neighbor in the case of the left border or right neighbor in the case of the right border) is part of the list. In case it is false, this cell is part of the clipping, if the value is bigger than the smallest value of the clipping list. A new value of the list is read (step 1).

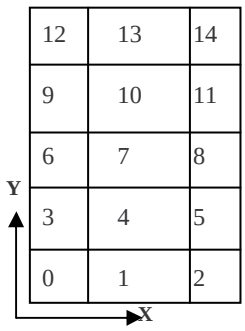


Figure 5. Cells identifiers of the structured model.

Obtaining the cells that are part of the clipping surface, the sub-volume is generated, using the other cells of the volume that are in reason of depth in z. With the clipping, it is possible to analyze the oceanic information through interaction techniques and animation.

5. CONCLUSIONS AND FUTURE WORK

The curvilinear clipping tool allows selecting volume regions in a more accurate way through the manipulation of splines to generate the clipping curve. The developed prototype introduces a new approach to select a 3D ROI, provides an animation tool and an interactive scalar and vector field visualization for NetCDF data, very common in such kind of tools. All the tools were developed in order to support the work of the CENPES oceanographers. Now we are interested in the evaluation of the developed tools through the elaboration and application of tests with users. Our next topics of investigation are: development of new visualization techniques, as isosurface and streamlines, and new readers to other data files format.

6. REFERENCES

[Bar87a] Bartels, Richard H. An introduction to Splines for use in Computer Graphics and geometric modeling. Los Altos, CA : M. Kaufmann, 1987.

[Cdf05a] CDF. CDF Homepage, Available at: nssdc.gsfc.nasa.gov/cdf/cdf_home.html.(June, 2005).

[Cro01a] Crossno, P.; Angel, E.; Munich, D. Case Study: Visualizing Ocean Currents With Color and Dithering. In: IEEE symposium on parallel and large-data visualization and graphics, 2001. Proceedings… (2001).

[Ebe06a] D.S. Ebert, P. Rheingans. Visualization Techniques. Available at http://www.cs.umbc.edu/~rheingan/691V/aug99/index.html. (May, 2006).

[Gai02a] Gaither, K.; Moorhead, R. J. Visualizing Vector Information in Ocean Environments. In: IEEE Visualization, 2002. Proceedings... IEEE Computer Society, (2002). p. 529 – 532.

[Hea97a] Head, M. E. M. et al. Applications of 3D visualizations of oceanographic data bases. IEEE/ MTS OCEANS, New York, v. 2, n. 6-9, p. 1210 – 1215, (1997).

[Koc84a] Kochanek, D.; Bartels, R. Interpolating Splines with Local Tension, Continuity, and Bias Control. Computer Graphics, v. 18, n. 3, p. 33-41, (July 1984).

[Mel03a] Mellor, G. L. Users guide for a three-dimensional, primitive equation, numerical ocean model. 53 p. Relatório Técnico-Programa de Ciências Atmosféricas e Oceânicas, Universidade de Princeton, (June, 2003).

[Nat06a] The National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign. The NCSA HDF Home Page. Available at http://hdf.ncsa.uiuc.edu/. (May, 2006).

[Pol04a] Polito, S. Paulo. Oceanografia Dinâmica. 71 p. Apostila. Departamento de Oceanografia Física, Instituto Oceanográfico da Universidade de São Paulo, São Paulo, (2004).

[Rew90a] Rew, R.; Davis, G. NetCDF: an interface for scientific data access. IEEE Computer Graphics and Applications, Los Alamitos, v. 10 , n. 4 , p. 76–82, (July, 1990).

[Ros89a] Rosenblum, L.J. Visualization of Experimental Data at the Naval Research Laboratory. US Naval Res. Lab., Washington, DC, (1989).

[Uni05a] Unidata. Unidata Homepage. Available at: http://my.unidata.ucar.edu/content/software/netcdf/index.html. (June, 2005).

Computer composition of transformed classical surfaces as the ways and means for constructing visual models of realistic objects (New software application “ArtMathGraph”)

Galina S. Rachkovskaya
107 Pushkinskaya Str. # 25
344007, Rostov-on-Don, Russia
galina.rachkovskaya@ verizon.net

Yuriy N. Kharabayev
1Б Budennovskiy Ave. # 46
344008, Rostov-on-Don, Russia
kharabayev@aaanet.ru

Natalya S. Rachkovskaya
1006 Moorefield Hill Pl
22180, Vienna, VA, USA
natashar@verizon.net

ABSTRACT

The new software application “ArtMathGraph” (AMG) is developed for constructing a visual representation of predetermined complicated geometrical forms as visual models of realistic objects, including objects of Nature. This application is based on the methods of Analytical Geometry and Computer Graphics only.

The AMG application includes two basic methodical parts: 1 - the analytical modeling and computer visualization of multiple different sets of new mathematical transformed surfaces as components for a consequential computer superposition of these surfaces; 2 - interactive visual modeling of predetermined geometrical forms by using the computer superposition of constructed mathematical transformed surfaces. By experience, consecutive use of all means of the AMG application allows obtaining various compositions of visual images of realistic objects (Figures 1, 2).

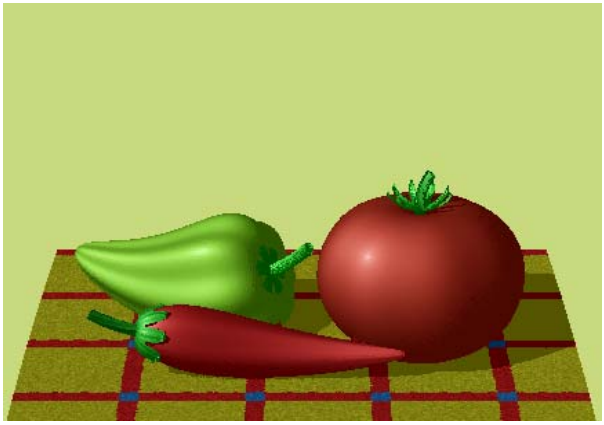


Figure 1

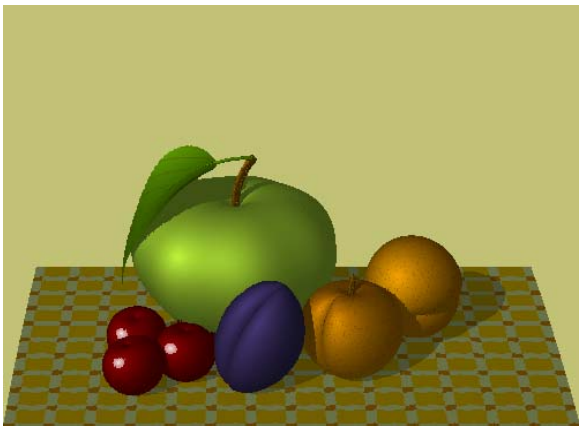


Figure 2

Keywords

Computer graphics, analytical modeling, visualization.

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

1. INTRODUCTION

The problems in constructing visual models of realistic objects, including objects of Nature, can be solved by the means of computer compositions of mathematically transformed classical surfaces. This solution is implemented in the new software application “ArtMathGraph” (AMG). The AMG application is developed as a computer mean for constructing visual images of complicated geometrical forms as models of realistic objects.

The AMG application includes:

1) basic set of initial well-known classical curves and surfaces;
2) some groups of mathematical transformations;
3) means for computer construction and visualization of the mathematical transformed surfaces;
4) means for interactive visual mathematical modeling of predetermined complicated geometrical forms as visual models of realistic objects;
5) means for obtaining various compositions of visual images of realistic objects.

2. ANALYTICAL MODELING AND COMPUTER VISUALIZATION OF MATHEMATICAL TRANSFORMED CLASSICAL SURFACES

The mathematical transformed surfaces are a result of applying the mathematical transformations to the analytical representations of initial well-known classical surfaces such as plane, cone, cylinder, sphere, ellipsoid, etc.

It is obvious that it is possible to construct multiple different sets of new mathematical transformed surfaces by combining the available values of variable parameters in the equations of Initial Curves (or Initial Surfaces) and/or of mathematical transformations. The analytical statement and software application for constructing and computer visualizing a large number of new mathematical transformed surfaces was developed previously based on conical transformations of torses as examples of Initial Surfaces [1-4]. It should be noted that the surfaces obtained using this software application are displayed on a screen in a 'frame' or 'solid' style [3]. Surfaces in the "solid" style are displayed in a user-selected color using variable color brightness to create a 3D effect. As example, the "frame" and the "solid" styles in the AMG application are shown in the Figures 3, 4.

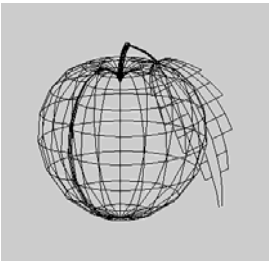


Figure 3

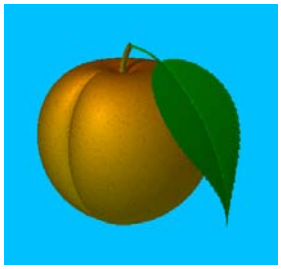


Figure 4

2.1. Analytical Representation of Initial Surfaces

Analytically the Initial Surfaces are represented by parametric equations:

$$x = x(u,v); \quad y = y(u,v); \quad z = z(u,v).$$

The Initial Surface equations are described in common form with use of their typical parameters. The selection of values for these parameters is available to the user and allows performing some deformations of the surfaces. On the whole, it is extension and compression.

The more significant surface modifications are achieved with the use of special transformations. There are two groups of mathematical transformations (the group "*Before*" and the group "*After*") differed in their impact on the analytical representation of surfaces.

2.2. Mathematical Transformation Groups "Before" and "After"

The mathematical transformation group "*Before*" is used for equations of surfaces on the level of arguments-parameters *u* and *v*; in essence, they change the analytical representation of surfaces.

With the help of these mathematical transformations the smooth surfaces can be performed cut, wavy, or lobulated. Actually, transformations of the group "*Before*" extend a basis set of Initial Surfaces. It is possible, for example, to derive a pyramid or prism by using cutting for a cone or cylinder accordingly. Mathematical transformations of the group "*After*" are performed for the values *x, y, z* calculated by proceeding from parametrical equations subject to transformations of the group "*Before*". The group "*After*" contains mathematical transformations of the surfaces twisting about axis *oz* (which is usually the axis of rotation for surfaces of rotation) and bending along the arc of a circle in different directions.

2.3. Mathematical Transformed Cone and Cylinder Surfaces (Analytical Representation and Computer Visualization)

For methodical example, the result from some mathematical transformations of Cone or Cylinder Surfaces as initial surfaces is shown in the Figures 5-16.

Fig. 5 Fig. 6 Fig. 7 Fig. 8 Fig. 9 Fig. 10

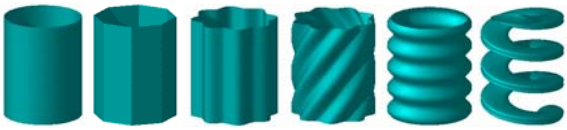


Fig. 11 Fig. 12 Fig. 13 Fig. 14 Fig. 15 Fig. 16

There are parametric equations of the Initial Cone and Cylinder Surfaces (Figures 5 and 11) and the Transformed Cone Surfaces (Figures 6-10) as results

of applying the mathematical transformations of groups "*Before*" and "A*fter*" to the Initial Cone Surface cited below. (The parametric equations of Transformed Cylinder Surfaces and Transformed Cone Surfaces are analogous.)

Initial Cone Surface (Fig. 7):

$$x = av\cos u;$$
$$y = av\sin u;$$
$$z = v; \qquad (1)$$
$$0 < u \le 2\pi,\ 0 \le v \le h.$$

Initial Cylinder Surface (Fig. 13):

$$x = a\cos u;$$
$$y = a\sin u;$$
$$z = v; \qquad (2)$$
$$0 \le u \le 2\pi,\ 0 \le v \le h.$$

Transformed Cone Surfaces (Fig. 8-12):

1. Cutting (Fig. 8) – group "*Before*"

$$x = a\frac{\cos\frac{\pi}{k}\cdot\cos u}{\cos(\frac{\pi}{k}-\varphi)}v;$$
$$y = a\frac{\cos\frac{\pi}{k}\cdot\sin u}{\cos(\frac{\pi}{k}-\varphi)}v;$$
$$z = v; \qquad (3)$$
$$\varphi = u - [\frac{k}{2\pi}\cdot u]\frac{2\pi}{k},$$
$$0 \le u \le 2\pi, \quad 0 \le v \le h.$$

k –quantity of sides.

2. Sinus-waviness (Fig. 9) – group "*Before*"

$$x = av(1 + d\sin ku)\cos u;$$
$$y = av(1 + d\sin ku)\sin u;$$
$$z = v; \qquad (4)$$
$$0 < u \le 2\pi,\ 0 \le v \le h,$$

k –quantity of waves, d –height of a wave.

3. Quasi-epicycloidal lobularity with twisting along axis *oz* (Fig. 10) – group ("*Before*" + "*After*")

$$X = x\cos\omega z + y\sin\omega z;$$
$$Y = -x\sin\omega z + y\cos\omega z;$$
$$Z = z, \qquad (5)$$

ω – determines direction and degree of twisting,
x, *y*, *z* are calculated according to the equations of quasi-epicycloidal lobularity – group "*Before*"

$$x = av((1+k)\cos u - \cos((1+k)u));$$
$$y = av((1+k)\sin u - \sin((1+k)u));$$
$$z = v; \qquad (6)$$
$$0 < u \le 2\pi,\ 0 \le v \le h,$$

k –quantity of lobes.

4. Thickening (Fig. 11) – group "*After*"

$$X = \frac{x}{1 + p(v - v_c)^2};$$
$$Y = \frac{y}{1 + p(v - v_c)^2};$$
$$Z = z, \qquad (7)$$

x, *y*, *z* are calculated according to the equations (1),
p – determines degree of thickening,
v_c – determines position of thickening along axis *oz*.

5. Bending along axis *oy* (Fig. 12) – group "*After*"

$$X = q - (q - x)\cos\frac{z}{q};$$
$$Y = y;$$
$$Z = (q - x)\sin\frac{z}{q}, \qquad (8)$$

x, *y*, *z* are calculated according to the equations (1),
q - determines direction and degree of bending.

3. MEANS OF INTERACTIVE VISUAL MATHEMATICAL MODELING OF PREDETERMINED COMPLICATED GEOMETRICAL FORMS

The AMG application includes the means of interactive visual mathematical modeling of predetermined complicated geometrical forms. It allows the user, guided by a screen view, creating visual images, adjusting parameters, as well as deriving realistic three-dimensional images. During the process of interactive visual mathematical modeling the user can change object's location and color, background color, and direction of light.

Consecutive mathematical transformations of the Initial Surfaces and computer superposition of the constructed transformed surfaces allows obtaining visual images of Complicated Geometrical Forms as models of various realistic objects (Fig. 1, 2, 17-19).

Figure 17

Figure 18

Figure 19

 ISBN 978-80-86943-99-2

4. MEANS OF OBTAINING ARTISTIC COMPOSITIONS OF VISUAL IMAGES

The following operations are implemented in the AMG application for obtaining artistic compositions:

- constructing visual images of individual geometrical forms as models of realistic objects;
- combining individual images into a composition of derivable visual images;
- arranging each individual image and a composition of individual images in the space with its 3D-representation on the screen;
- choosing color for individual images;
- choosing direction of light and constructing corresponding distribution of light and shade;
- considering perspective as a system of representation of visual images on the screen.

Several "Mathematical Pictures" are shown in the Figures 1, 2, 17-22 as some illustrations of the AMG application possibilities.

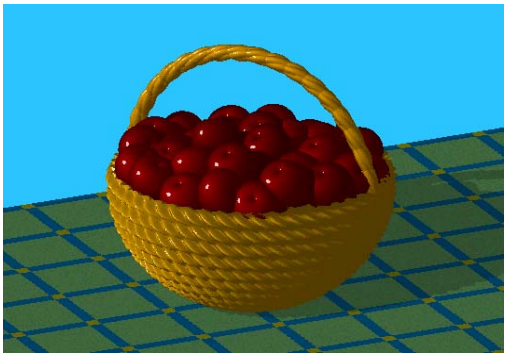


Figure 20

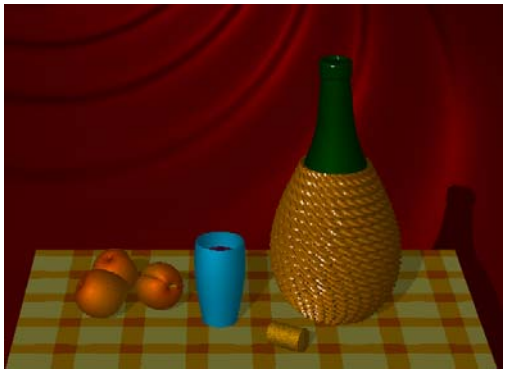


Figure 21

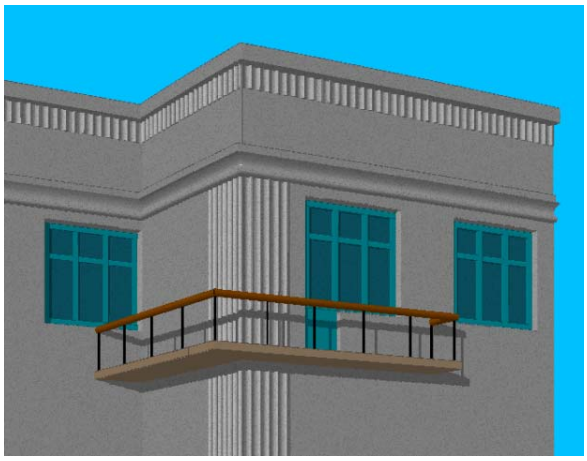


Figure 22

By experience, consecutive use of all means of the AMG application allows obtaining various artistic compositions of visual images of realistic objects by the methods of computer superposition of the mathematical transformed classical surfaces.

REFERENCES

[Con02a] Rachkovskaya, G.S. and Kharabayev, Yu.N. Mathematical modelling of kinematics of ruled surfaces based on conical transformations of torses. Proceedings of the 10th International Conference on Geometry and Graphics. Kyiv, Ukraine, Vol.1, pp. 283-286, 2002.

[Con02b] Rachkovskaya, G.S. and Kharabayev, Yu.N. The computer graphics of modelling of kinematic linear surfaces based on rolling a cone along a torse. Proceedings of the 12th International Conference GraphiCon' 2002, N.Novgorod, Russia, pp. 153-156, 2002.

[Con04a] Rachkovskaya, G.S., Kharabayev, Yu.N., and Rachkovskaya, N.S. Computer graphics of kinematic surfaces. Proceedings of the 12-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2004, Plzen, Czech Republic, p.p. 141-144, 2004.

[Con04a] Rachkovskaya, G.S., Kharabayev, Yu.N., and Rachkovskaya, N.S. The computer modelling of kinematic linear surfaces (based on the complex moving a cone along a torse). Proceedings of the International Conference on Computing, Communications and Control Technologies, Austin (Texas), USA, Vol.1, p.p. 107-111, 2004.

POSTER:Development of Virtual Goldfish Scooping Using Haptic Device

Ken-ichi Kobori Takanori Hori Norihiro Nakamura Koji Nishio
kobori@is.oit.ac.jp hori@ggl.is.oit.ac.jp nakamura@is.oit.ac.jp nishio@is.oit.ac.jp

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

ABSTRACT

Goldfish scooping is one of the traditional Japanese play. In general, people scoop goldfish in a shallow water tank using a fragile water-thin dipper. In this paper, we describe virtual goldfish scooping system using haptic device and stereoscopic display.

Keywords

Virtual reality, Haptic device, Stereoscopy, CG

1. Introduction

A virtual reality technology has been in the spotlight, because the capability for processing three-dimensional CG has been increasing in recent years. Many studies in which traditional Japanese culture is realized in a virtual environment have been proposed [Ter01a], [Oka01a]. We took notice of goldfish scooping. It is one of the traditional amusement in summer festival. We have developed virtual goldfish scooping system to evaluate the difference between real goldfish scooping and virtual goldfish scooping. The sense of touch and real time display are very important to achieve virtual goldfish scooping system. A haptic device is used to get the sense of touch. We use a stereoscopic display to represent three dimensional objects such as goldfishes. In addition, we achieve natural motion of goldfishes by using the method of crowd animation [Kay01a].

2. Virtual goldfish Scooping

2.1 System configuration

The system consists of three hardware components, a three dimensional input device PHANTOM[Sen01a] made by SensAble Technologies, a Intel Xeon 2.4GHz×2, and a 17inches stereoscopic display. Figure 1 shows our system.

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

Figure 1: Setup of virtual goldfish scooping

2.2 Motion of goldfish

Each goldfish object has a circle area called personal space to avoid other fishes or obstructions when they approach. The position of a goldfish is the center point of its personal space as shown in Figure 2.

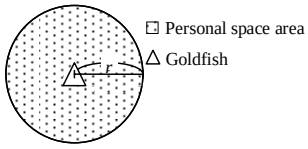


Figure 2: Personal space

A goldfish avoids other fishes or obstructions using an avoidance vector as shown in Figure 3 when they break into its personal space area . First, goldfish ‘A ‘checks whether other fishes exist inside of the personal space of ‘A’ or not. Fish ‘B’ is an invader

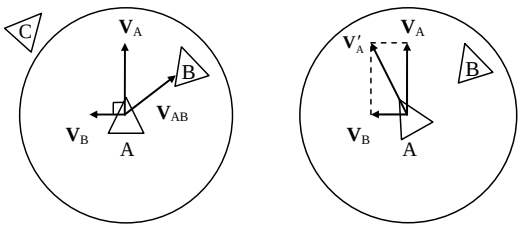
in Figure 3. Next, we calculate affect vector $\mathbf{V}_B$ that 'B' has influence on 'A'. The direction of affect vector $\mathbf{V}_B$ is perpendicular to $\mathbf{V}_A$ which is a velocity of gold fish 'A' when 'B' exists on the right side of 'A' shown in Figure 3(a). The length of $\mathbf{V}_B$ is given by equation (1). As the distance between 'A' and 'B' becomes short, the magnitude of $\mathbf{V}_B$ grows large. As a result, 'B' has big influence on the way of 'A'.

$$|\mathbf{V}_B| = |\mathbf{V}_A|\left(1-\left(|\mathbf{V}_{AB}|/r\right)\right) \quad (1)$$

$\mathbf{V}_{AB}$: Vector from position 'A' to 'B'

r:Radius of personal space

Composite vector $\mathbf{V}_A$' is calculated shown in Figure 3(b). Fish 'A' turns in the direction of $\mathbf{V}_A$'.



(a) (b)

Figure 3: Generation of avoidance vector

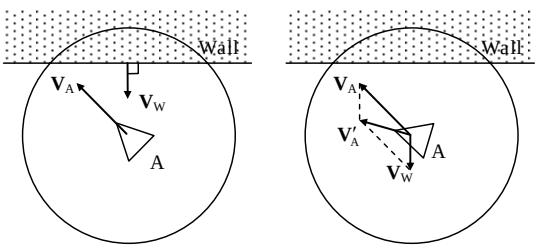
This process is also applied in the case that a fragile water-thin dipper invades the inside of the personal space area.

When fish 'A' gets close to a wall of the tank , affect vector $\mathbf{V}_W$ perpendicular to the wall is generated as shown in Figure 4(a). The length of $\mathbf{V}_W$ is given by equation (2).

$$|\mathbf{V}_W| = |\mathbf{V}_A|\left(1-\left(dist/r\right)\right) \quad (2)$$

dist : Distance between fish 'A' and the wall

Avoidance vector $\mathbf{V}_A$' is calculated shown in Figure 4(b).



(a) (b)

Figure 4: Affect vector V_W from the wall

2.3 Thin dipper

The thin dipper breaks easily because it is made of thin paper. We achieve this process that the paper of thin dipper breaks using the following two factors.

1) The damage that thin dipper sustains under the water.

2) The damage that thin dipper sustains when goldfish is caught.

The thin dipper has the power of endurance and the power reduces according to these damages. The paper of the dipper breaks when its power runs out.

2.3.1 The damage that thin dipper sustains under the water

The damage D_W under the water is given by equation (3)

$$D_W = s_w\left(\mathbf{V}_S \cdot \mathbf{V}_N\right) \quad (3)$$

where $\mathbf{V}_S$ is a velocity of the dipper and $\mathbf{V}_N$ is normal vector of the dipper shown in Figure 5. s_w is a damage coefficient under the water.

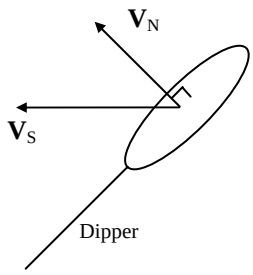


Figure 5: Relation between $\mathbf{V}_S$ and $\mathbf{V}_N$

2.3.2 The damage in catching goldfish

The damage depends on the position on the dipper where goldfish is caught by. The damage is biggest when the fish is caught on the center of the dipper.

The damage D_S is given by equation (4)

$$D_S = s_k(1-(len/r)) \quad (4)$$

Where s_k is a damage coefficient in catching goldfish and r is the radius of the dipper. *len* is the length between the center of the dipper and the position of the caught fish shown in Figure 6.

 ISBN 978-80-86943-99-2

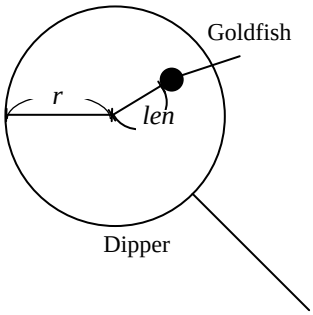


Figure 6: Position of goldfish on the dipper

2.4 Sense of touch

We have achieved the following the touch of sense.

1) Water resistance

2) Sense of touch when a goldfish is caught

2.4.1 Water resistance

Water resistance is caused when we move the dipper under the water. Water resistance F[N] is given by equation (5)

$$\mathbf{F} = k\left(\frac{\mathbf{V}_S \cdot \mathbf{V}_N}{|\mathbf{V}_S||\mathbf{V}_N|}\right)\overline{\mathbf{V}}_S \qquad (5)$$

where k is water resistance coefficient and is set to 0.3 in this experiment. $\overline{\mathbf{V}}_S$ is the opposite vector of velocity of dipper. Water resistance depends on an inclination of the dipper under the water.

2.4.2 Sense of touch when goldfish is caught

The caught goldfish jumps up and down on the dipper. The arm of PHNTOM vibrates up and down to achieve the feeling of jumping fish. The force F_y is given by equation (6)

$$\mathbf{F}_y = a\sin(t \times x) \qquad (6)$$

where a is amplitude of vibration and x is the number of vibration. t is the elapsed time after a fish is caught. The value of a and x depend on the size of a goldfish. The amplitude of vibration is in proportion to the size of a goldfish and the number of vibration is in inverse proportion to it.

2.5 Animation of goldfish

Sight information is important to improve reality of the motion. A goldfish swims while bending the body. In order to realize animation of goldfish, the system displays five images shown in Figure 7 from the left to the right in turn.

Figure 7: Five images to animate goldfish

2.6 Execution of the system

Figure 8 shows an example of CG image on the screen. The portion of paper is broken because of the damage from water.

Figure 8: Screen image of the system (without stereoscopy)

3. Experimental results

We experimented on a subjective judgment to verify the following 6 items of the system and evaluated these items by five grades (5: excellent,….,1: poor). In the subjective judgment, total average score for 13 test subjects was 4.3.

Evaluation items and the average score are as follows;

1) Motion of goldfish: 4.7

2) Handling feeling of the dipper under the water: 4.4

3) Sense of touch when goldfish is caught: 4.1

4) Realty of breaking paper dipper: 3.5

5) CG image quality by stereoscopic display: 4.3

6) Performance of real-time: 4.8

Generally, each score is high. However, the score in item 4) is slightly low. Several test subjects reported that they could not recognize when the paper of the

 ISBN 978-80-86943-99-2

dipper tore. Figure 9 shows the state of the experiment.

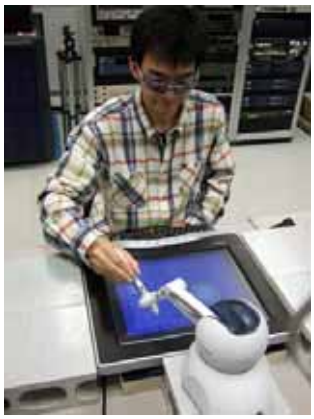


Figure 9: Experimental picture

4. Conclusions and future work

We proposed virtual goldfish scooping system using haptic device and stereoscopic display. Experimental results show that proposed system is same as the real goldfish scooping as for the sense of touch and reality of the goldfish. The score of item 4) is lower than the total score. We intend to improve the process of breaking paper dipper. We are also working on the improvement in representation of thin paper of dipper to imitate to real dipper.

References

[Ter01a]Terai R., Nakamura N., Nishio K., and Kobori K. Virtual India Ink Painting System by Using Haptic Device, IEICE conf.proc, 2005

[Oka01a]Okada M., Mizuno S., Toriwaki J. Virtual Sculpting and Virtual Woodblock Printing by a Model-Driven Scheme, The Journal of the Society for Art and Science, pp.74-84,Vol.1,No.2,2002

[Sen01a]SenAble, http://www.sensable.com/

[Kay01a]Kayahara K., Nishio K., and Kobori K. Crowd Behavier Animation in Three-Dimensional Space, The Journal of the institute of image information and television engineers, pp.68-74,Vol.58,No.4, 2004

 ISBN 978-80-86943-99-2

Poster: Design Patterns for Multithreaded Software Pipelines in Real Time Applications

Stefan Preuss
University of Karlsruhe, IBDS
Am Fasanengarten 5
76128, Karlsruhe, Germany
stpreuss@ira.ika.de

Alfred A. Schmitt
University of Karlsruhe, IBDS
Am Fasanengarten 5
76128, Karlsruhe, Germany
aschmitt@ira.ika.de

ABSTRACT

This paper presents design patterns that will help in the task of parallelizing graphical real time algorithms, according to the example of a visual real time 3D reconstruction algorithm. These algorithms can often be designed as a dataflow graph, so they can be coarsely granular parallelized in a pipeline pattern. With these patterns, the design process of the parallelization is detached from the design of the graphical algorithm. The advantages and drawbacks of these patterns are discussed with regard to speed, but also to handling and error-proneness and the demanded robustness of real time applications, due to the varying workload of the different steps or data loss or obsoleteness during processing.

Keywords

parallelization, real time application, software architecture.

1. INTRODUCTION

Decent workstations are often based on SMP systems and recently even multicore processors are becoming affordable. One great advantage of these UMA designs is the low cost of communicating complex data structures (i.e. 3D meshes) between threads. Many graphical algorithms can be designed as processing pipelines similar to dataflow oriented processors and thus be parallelized coarsely granular, in order to get a faster response and/or higher output frequency of results. The pipeline approach results mostly (but not exclusivly) in the latter.

The danger of such an attempt arises from the indeterministic process flow. Deadlocks, random data order, changing or deletion of data in use by other elements must be considered. Finding the resulting errors is hard and time consuming. The use of the presented design patterns will help to minimize these dangers, that occur from parallelizing an algorithm by giving each thread a clearly defined area of responsibility.

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

2. MOTIVATION

The pipeline parallelization patterns resulted from the parallelization of our reconstruction algorithm. It is based on the works of [Lau94] and [Fau03]. [Lau94] introduced the concept of a visual hull and, based on this, [Fau03] developed his automatic reconstruction process of real objects with a camera on a robot arm that uses a new variant of the volume intersection approach. Due to the high performance of this algorithm, we try to use it for markerless VR immersion in a real time environment. The adaptive background model for registering the silhouettes is described in [Thu05]. The real time vertex reduction process of this project is described in [Piz06]. A rival approach [Fra04] is however based on *Bulk Synchronous Programming Scheme* by [Val90]

3. BASIC PIPELINE DESIGNS

The design of a pipeline has to consider four different aspects: the arrangement of the processed data, the communication channels between the pipeline steps, the connection topology and the controlling of the working threads.

The basic patterns presented in this chapter were inspired by [Joh04], who gives an overview of the first achievements in the nowadays disregarded dataflow processors that led to the visual dataflow programming 'languages' or tools and describes similar problems occurring during the design of a dataflow processor or a dataflow oriented software.

Another fundamental book is [Gam95], a handbook on patterns for object oriented software design. These patterns turn out to be helpful for organizing thread responsibilities as well.

The different patterns for each aspect generally describe a trade-off between flexibility, performance, parallel distribution and error proneness.

Connection Topology

Contrary to a normal dataflow graph, it is a more flexible approach to let each step decide for itself which type of data it will accept and send its results indifferently to its successors. The connection topology defines in which succeeding steps the data has to be buffered.

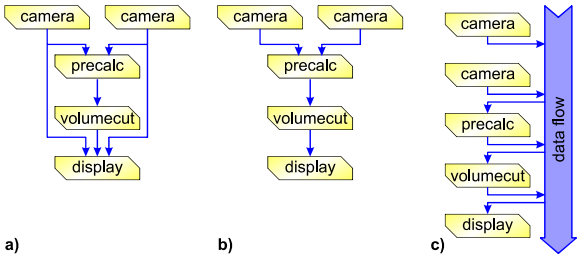


Figure 1: Different pipeline arrangements: a) typical data flow graph. b) assembly line arrangement. c) dataflow arrangement.

As shown in Figure 1, the *assembly line arrangement* is a strictly pipeline approach. As the data pass through each step of the pipeline, it has to be buffered in each of them. Unneeded data is forwarded unprocessed to its successor. In the *dataflow arrangement*, the different steps pick and buffer only their demanded data out of the data stream. The results are sent into the dataflow.

Data Arrangements

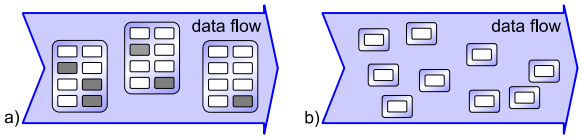


Figure 2: Different handling of the data pieces: as one complete set in the workpiece arrangement a), or free floating single data chunks b).

As shown in Fugure 2, in the *workpiece* arrangement all data is packed into one container, so that each pipeline step works on a complete data set. This arrangement might not exploit the entire potential of parallelization, but handling and overviewing is easy.

In the *free data flow* arrangement, the different types of data are sent independently through the pipeline. The pipeline steps could start working on the first incoming data part but have to take care to gather all their required data. At a first glance, this distribution might be a good idea but the effort to pick and gather all the needed data might sap all performance gain.

Communication Arrangements

First, the event that triggers a communication between the pipeline steps has to be defined. In the *data driven* concept, the steps of the pipeline become active and produce results on incoming data. In the *demand driven* concept, a step reacts on a request for its result. Whereas the first is the most used concept for data processing tasks, the latter is useful for interactive tasks.

The possibilities to layout the communication channel between the pipeline steps differ as well. The most simple way is the *newest-buffered* communication, where only the latest data is present in the channel and older data is lost. Different buffers for each type of data have to be established. This is usually the choice for visual real time applications.

Another communication arrangement is the *synchronous* communication. The channel is locked for new input until the old data is picked up by the succeeding step. This data arrangement assures that there is no data loss, but it could slow down the pipeline, or even reserialize the parallelization.

The last arrangement is the *buffering-all* construct. The sent data is parked into a buffer structure until the receiving step picks up the data for processing. Although different buffer techniques are concernable (e.g. stack, tree), in most cases a FIFO queue will be adequate. This arrangement accommodates the possibly fluctuating processing workload of the different pipeline steps. The data will not be lost and the pipeline steps do not block each other.

Thread control

active waiting	conditional waiting
DO **IF** data is waiting do work reduce sleeptime **ELSE** increase sleeptime sleep(sleeptime) **UNTIL** should end	**DO** wait for data do work **UNTIL** should end

Table 1: The working loops of an active or conditional waiting thread.

A simple pseudocode illustration of the two different thread loop designs is shown in Table 1. The *active waiting* thread repeatedly checks in its processing loop wether there is any data waiting to be processed. The thread could be put into a temporary dormant state to reduce unnecessary processing cycles. The sleeping time can increase with every unnecessary

check. The *conditional waiting* thread is initially in a waiting state and only activated if needed.

The advantage of the *conditional waiting* thread is that processor time is only used when there is effective work to do, and the thread is immediately reacting. On the other hand the *active waiting* thread is independent from the overall arrangement. All possible circumstances are local in the responsibilities of this certain thread, so the danger of deadlocks and race conditions is minimized. It is suitable for slow reacting tasks, such as user interface (e.g. rendering 24 times per second).

4. APPLIED MULTITHREADED PIPELINE DESIGN

Our reconstruction environment has typical real time conditions and requirements. There are different data types to be processed and each set of them is related to one point in time. A crucial challenge is that data parts may be missing or be obsolete due to newer data sets. Additionally, the different steps of the reconstruction process should be able to be dynamically activated during experimentation.

Data Container

Different types of data are encapsulated in a container object that holds, among other administrative data, a flag to mark if it is the last data piece with the given attributes. Usually the creating object is responsible for cleaning up its constructed data in order to avoid memory leaks during execution. To prevent complexity and bottlenecks as well as memory leaks and colliding access, we hold a strict policy: a data container and its data is accessed and in the responsibility of only one object in the pipeline at a time.

Managing the Waiting Queues

As data of different timestamps and types arrives at a pipeline step, a number of waiting queues must be handled for each communication channel between the steps. Thereby one exclusivly accessed queue collects the data of a specific timestamp. It is marked to be *full* if it received a container with a “last-one mark” for every expected data type. The queue is marked to be *done* if for every expected data type a container with the “last-one mark” was given out.

The different queues are managed by a factory construct (see Figure 3). This construct was not chosen for the typical reasons as described in [Gam95], but to create a central object to keep track of the access on its queues, similar to an inverse semaphore. So the access to a queue is monitored by a check in/ check out mechanism. This allows parallel access to different waiting queues. The competing access to one queue is managed by itself. Returned queues that are *done* and not checked out by other threads will be deleted. Queues that are older than this queue can be marked as being *done*, too. This eliminates obsolete or uncomplete queues.

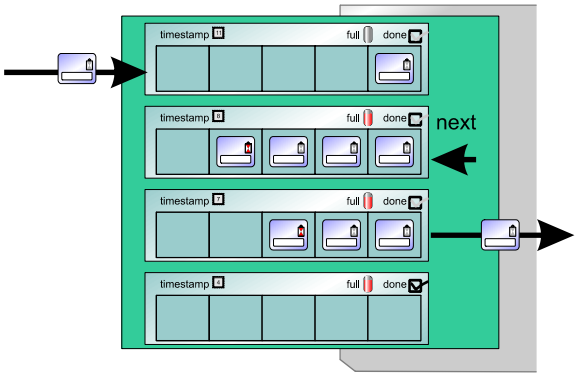


Figure 3: The queue factory for managing the different waiting queues for each timestamp.

One interesting aspect is the strategy how to determine a new queue for the receiving pipeline step after it has finished one (the next marked queue in Figure 3).

The different strategies of handling the queues resemble different aspects of the *newest-buffered* and *buffering-all* FIFO arrangements used here on complete data sets. A *synchronous* arrangement would be contradictory to the concept of a queue to uncouple sending and receiving steps. As data can be lost, and the pipeline step might wait unnecessary in a queue, it is possible to only switch to *full* queues. A compromise to reduce the waiting time may be to give priority to full queues and start processing non-*full* ones if no full one is present. Older queues are discarded and newer, but non-*full* queues are kept to be filled in the future.

In the experiments, it is well tried to keep track of the waiting full queues. Reaching a given threshold indicates that the receiving pipeline step is waiting for data that will never come, or is getting not enough processor time for its task. In the first case, the step has to discard its work on this timestamp. In the latter case, the preceeding steps have to be slowed down similar to the *synchronous* communication arrangement.

Pipeline Steps

The functional code of the pipeline step is encapsulated in different agents (similiar to the visitor pattern described in [Gam95]) to allow quick allocation of processing steps to a working thread during experimentation. As can be seen in Figure 4, each step consists of a queue management factory for the incoming data, one or more agents to process the data and a thread running through all of them. A filter might be added in front of the queue management factory to send the unprocessed data types to the succeeding steps immediately. This filter

transforms an *assembly line arrangement* into a *dataflow arrangement*.

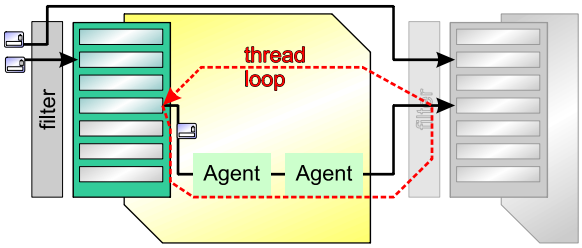


Figure 4: The pipeline step. It capsulates its incoming communication channel, its functional entities (agents) and the working thread.

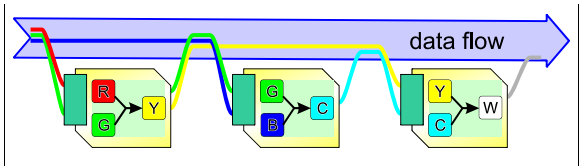
Special Pipeline Steps

The *gatherer* is a pipeline step to manage incoming data of several preceeding sending steps and set the "last-one" mark on the last container of all sending predecessors. The *distributor* has to copy the incoming data for several succeeding receiving steps.

This functionality can be implemented in a standard pipeline step but, as not every step has several predecessors or successors, there is no need for this costly administration effort.

5. THE IDLE LOAD EFFECT

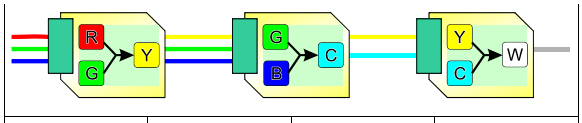
At a first glance, a *dataflow arrangement* with one of the *newest* strategies seems to distribute the work best among the pipeline steps, while minimizing administrative work.



timestamp	R+G > Y	G+R > C	Y+C > W
1	abort (no R)	-	-
2	success	abort (no B)	abort (no C)
3	abort (no R)	abort (no G)	-
4	success	abort (no B)	abort (no C)
5	abort (no R)	-	-
6	success	success	success

Table 2: Example of the idle load effect. The data flow arrangement has to abort a lot of started processing work due to missing data.

In the following example there is a *dataflow arranged* pipeline with *newest* strategy (see Table 2) in competition with a *assembly line* pipeline with *newest-full* strategy (see Table 3). There are three data sources sending data types: Red, Green, and Blue. The unreliable Red and Blue data sources shall send their data every 2nd, respectively 3rd timestamp. The three-step pipeline processes Red, Green and Blue data to the desired White data type. As can be seen easily, it can provide White data only in every 6th timestamp. But here the assembly line arrangement waits in the first step for a full queue up to the 6th timestamp, thus filtering out incomplete data sets. The supposed faster first arrangement keeps the system busy with ten futile proccessing cycles. As part of the data might be lost due to hardware or algorithmic issues or by the *newest* strategies, the data loss and futile processing work increase.



timestamp	R+G > Y	G+R > C	Y+C > W
1	discard	-	-
2	discard	-	-
3	discard	-	-
4	discard	-	-
5	discard	-	-
6	success	success	success

Table 3: Example of the idle load effect. The assembly line arrangement also has to wait for a complete data set, but does not waste processor time in the meantime.

6. REFERENCES

[Fau03] Fautz, M. Objekt- und Texturrekonstruktion mit einer robotergeführten Kamera. Shaker Verlag, 2003.

[Fra04] Franco, J.S., Ménier, C., Boyer, E. and Raffin, B. A Distributed Approach for Real Time 3D Modeling. Proceedings of the 2004 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'04) IEEE. 2004.

[Gam95] Gamma, E., Helm, R., Johnson, R. and Vlissides, J. Design Patterns, Addision-Wesley Publishing Company, 1995.

[Joh04] Johnston, W.M., Hanna, J.R.P. and Millar R.J. Advances in Dataflow Programming Languages. ACM Computing Surveys, Vol. 36 No. 1, ACM Press 2004.

[Lau94] Laurentini, A. The visual hull concept for silhouette-based image understanding. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, N. 2, 1994.

[Piz06] Pizarro, F., Preuss, S. Simplification of Reconstructed Meshes in Real Time. CASA 2006 Proceedings, Computer Graphics Society (CGS)

[Thu05] Thüring, S., Herwig, J. and Schmitt, A. Silhouette-based Motion Capture for Interactive VR-Systems including a Rear Projection Screen. CASA 2005 Proceedings, Computer Graphics Society (CGS)

[Val90] Valiant, L.G. A Bridging Model for Parallel Computation. Communication of the ACM 33(8), 1990.

POSTER: Controlling 3D characters by hand motion

Misae Saitoh
Hosei University
Koganei Kajinocho 7-3-2
184-8584, Tokyo, Japan
saitohmisae@philos.k.hosei.ac.jp

Kenji Miyamoto
Hosei University
Koganei Kajinocho 7-3-2
184-8584, Tokyo, Japan
miyaken@hosei.ac.jp

ABSTRACT

A method to control motions of complex 3D characters interactively using hand movements is described. After capturing actor's motion, by motion capture systems transfer it as it is to 3D character's motion. Motion capture fails, however, when the target character is of shape far different from that of the actor captured. This article proposes manipulating 3D character by controllable parameters extracted from monocular image of the actor's hand without reconstructing its posture. Using Digital Puppet System which implements this method allows a user to easily manipulate characters of various shapes ranging from a human-type robot to a mollusk with large-scale set of joints. By extracting as many controllable parameters as possible, our method makes maximal use of available information which is not necessarily enough for posture reconstruction.

Keywords

3D character, hand motion, interactive animation, motion capture, model-based

1. INTRODUCTION

A method to control complex motions of 3D characters using hand movements is described.

For controlling 3D characters, motion capture first reconstructs the actor's posture from positional information of a set of markers and then, the postural parameters are used as they are to determine the character's posture. Recent researches [Cha05a] [Nor93a] [Sem98a] [Yin03a] aim to reduce the number of required markers. In general, motion capture has advantage in reproducing realistic motion.
However, the number of independent parameters that can be controlled than the accuracy of the motion capture is more important to move the character with connecting topology different from that of the human body. For example, to reconstruct hand posture, it needs a lot of markers because the hand has 19 joints, notwithstanding that actually controllable information is much less.

Digital Puppet System (DPS) doesn't reconstruct posture but extracts directly the feature parameter, a quantity related to posture, that can be controlled by the actor. More specifically, 1) DPS extracts feature parameters from the location information of the markers by model-based method [Att89a] [Kak00a] [Ued03a], and 2) uses these parameters as control parameters that determine the posture of the target character. In the actual system, watching user's hand with a marker attached to the tip of each finger, DPS estimates how much the fingers bend or stretch. Then, these feature parameters are assigned to the joints of the parts of the character, determining their bending angles. Here, a single parameter can possibly be used for more than one parts of 3D character.

Experiment showed that by using this system a user can control the movement of 3D character by his hand with rotating, moving and, bending or stretching its three fingers, Wriggle motions of a 3D character with multiple joints were also able to be controlled. Thus, whole body of 3D character was able to be controlled as if working on puppets.

Our method shows how to maximally make use of the available information, even if there is not enough information for reconstruction.

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

2. RELATED WORK

In methods other than motion capture, user moves sensors that can detect its own movement. [Las00a] controls the movement of the character by the direction and extent the mouse is moved. Video game machine Nintendo's Wii [Nin] controls the character by moving a remote control equipped with motion sensor. In these methods, a few parameters corresponding to the number of freedom could be

controlled independently but they are difficult to change concurrently.

3. EXTRACTING FEATURE PARAMETERS

Digital Puppet System directly extracts feature parameters without reconstructing actor's posture. Here, from the usability's point of view, it is preferable that the feature parameters can be controlled independently. From expressive power's point of view, it is preferable that as many parameters as possible are available given fixed number of markers. This chapter describes *model-based feature extraction* (*MBFX*) a model-based method to extract independent parameters from small number of markers attached to the tip of a finger tracked. Then, conversion of posture information on 3D character from feature parameter are also briefly described

3.1 Feature Extraction for Hand

3.1.1 2-D of hand model

The *feature model*, which relates feature parameters to quantities to be measured, is used to estimate *feature vector*, each element of which is a feature parameter, from the measured quantities. The proposed feature model, called *2-D hand model* (Fig. 1), is a model of a hand that bends and stretches three fingers, translates and rotates in the plane vertical to the optical axis of the camera. The model consists of a point P on the plane, and three lines which pass P and preserves their relative crossing angles, and points Q_1, Q_2 and Q_3 on these lines. The lines mean the directions of the fingers and Q_i means the tips of them. Based on this model, *feature vector* in 2-D hand model is represented by triple (p, γ, a). Here, $p = (p_x, p_y)$ is the two dimensional position of P, γ is the rotation angle from some initial value, and a is the vector whose elements are $|q_i|$ (i=1,2,3).

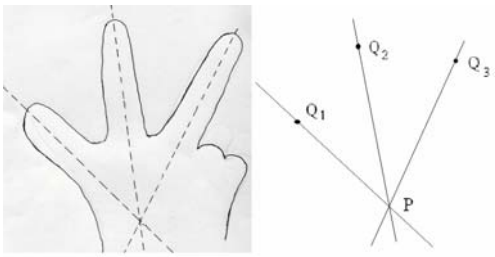


Figure 1. 2-D hand model

The next section describes the method to estimate, given an initial position of point P, the value of parameter (p, γ, a) changing at each time from observed position of points Q_1, Q_2, and Q_3.

3.1.2 Parameter Estimation

Initial value

Initial position P_0 of point P is assumed to be given. In the actual system, 2-D model is displayed on the screen. When three fingers fit to the lines, P_0 is obtained by regarding the displayed figure as the initial state of the model. Thus, the initial value is $(p, \gamma, a) = (0,0,|P_oQ_i|)$.

Estimation of feature vector

Change in (p, γ, a) according to change in markers' positions are estimated. The i-th marker's new position q_i' can be represented by their old position q as:

$$q_i' = a_i R_\theta u_i + \ell \ . \quad (i=1,2,3) \quad \cdot\cdot\cdot (1)$$

Here,

$$R_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}, \quad u_i = \frac{q_i}{|q_i|},$$

ℓ : displacement vector p'-p

Here, the unknowns in *eq*.(1) are a_i, R_θ and ℓ . Because the position changes continuously,

$$R_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \cong \begin{pmatrix} 1 & -\theta \\ \theta & 1 \end{pmatrix}$$

assuming θ<<1. Under this approximation, θ can be solved as:

$$\theta = b/c$$

$$b = u_{1y}u_{3y}u_{2x}q_{3x} - u_{1y}u_{2y}q_{2x}u_{3x}$$
$$-u_{1y}u_{3y}u_{2x}q_{1x} + u_{1y}q_{2y}u_{2x}u_{3x} + u_{1y}u_{2y}q_{1x}u_{3x} - u_{1y}q_{3y}u_{2x}u_{3x}$$
$$-u_{2y}u_{1x}u_{3y}q_{3x} + u_{3y}u_{2x}u_{1x}q_{1y} + u_{2y}q_{2x}u_{1x}u_{3y} + q_{3y}u_{2y}u_{1x}u_{3x}$$
$$-q_{1y}u_{2y}u_{1x}u_{3x} - q_{2y}u_{2x}u_{1x}u_{3y}$$

$$c = -u_{1y}u_{3y}u_{2x}q_{3y} - u_{1y}u_{3x}u_{2x}q_{3x} + u_{1y}u_{2x}q_{2x}u_{3x} + u_{1y}u_{3y}u_{2x}q_{1y}$$
$$-q_{1y}u_{1y}u_{3x}u_{2y} + u_{1y}q_{2y}u_{2y}u_{3x}$$
$$-u_{1x}q_{1x}u_{3x}u_{2y} + u_{3x}u_{2y}u_{1x}q_{3x} + q_{3y}u_{2y}u_{1x}u_{3y} + u_{3y}u_{2x}q_{1x}u_{1x}$$
$$-u_{1x}u_{3y}u_{2x}q_{2x} - u_{1x}u_{3y}q_{2y}u_{2y}$$

a_i, ℓ in *eq*. (1) is solved using this θ.

$$l_x = d/e, \ l_y = f/e$$

$$d = s_{2x}s_{1x}q_{1y} - s_{2x}s_{1y}q_{1x} - s_{2x}s_{1x}q_{2y} + s_{1x}s_{2y}q_{2x}$$

$$e = -s_{2x}s_{1y} + s_{1x}s_{2y}$$

$$f = s_{2y}s_{1x}q_{1y} - s_{2y}s_{1y}q_{1x} + s_{2y}q_{2x}s_{1y} - q_{2y}s_{2x}s_{1y}$$

$$a_1 = g/eq_1$$

$$a_2 = h/eq_2$$

$$a_3 = k/eq_3$$

$$g = s_{2y}q_{1x} - s_{2y}q_{2x} + s_{2x}q_{2y} - s_{2x}q_{1y}$$

$$h = -s_{1x}q_1 + s_{1y}q_{1x} + s_{1x}q_{2y} - q_{2x}s_{1y}$$

 ISBN 978-80-86943-99-2

$$k=-s_{2x}s_{1x}q_{1y}-s_{2x}s_{1y}q_{3x}+s_{2x}s_{1y}q_{1x}+s_{2x}s_{1x}q_{2y}+s_{1x}s_{2y}q_{3x}$$
$$-s_{1x}s_{2y}q_{2x}$$

Here, $R_{\theta}u_i = s_i$ is assumed.

After all, putting the above all together, we obtain the tracking procedure.

Tracking

1. $P := P_0$, $\gamma := 0$, $u_i := P_0 Q_i$;

2. Each time when the position of the markers change, repeat the following.

Obtain θ, ℓ, and a by calculation described above. Then

$p := p + \ell$;

$\gamma := \gamma + \theta$;

$a_i := |PQ_i|$;

3.2 Determining Target Posture

The remaining task is to transfer the parameter extracted above into posture information of 3D character in each frame. This is done simply by setting the value of bending parameter, latitude or longitude, of each joint of target to the value of some extracted parameter. Here, the structure of target 3D character is supposed as follows.

Posture information of 3D character

3D character consists of two or more parts, which are connected by joints. The parts form a tree structure with some part as its root and joints as its edges. Thus, parent-child relation naturally be introduced among parts. Using this structure, when the parent part moves, child part moves together in the coordinate of the parent part. The direction of flexing of a joint is represented by the latitude and the longitude and, thus, the entire posture of a 3D character is determined by flexing of all joints.

4. DIGITAL PUPPET SYSTEM

This chapter describes the overview of Digital Puppet System.

Fig. 2 shows the arrangement of the Digital Puppet System. The screen displays 3D character image together with the image of the user's hand. Actor controls 3D character while watching 3D character.

Figure 2. Digital Puppet System

Fig. 3 shows the screen part displaying the actor's hand with red, green and blue markers attached. Red, green, and blue markers are attached to the tip of the thumb, the forefinger, and the middle finger.

Figure 3. Hand with markers

Example 1

This example demonstrates manipulating 3D robot of the human type. Fig. 4 shows a screenshot during the operation. Acting person's thumb, forefinger, and middle finger correspond to the right arm, the left arm, and both feet of the 3D robot respectively. When the forefinger is bent, the right arm of 3D character is lowered, and is heighten when stretched. If the hand is rotated, the head of 3D robot slants, and if the hand is slid, the whole body of 3D robot moves in the same direction.

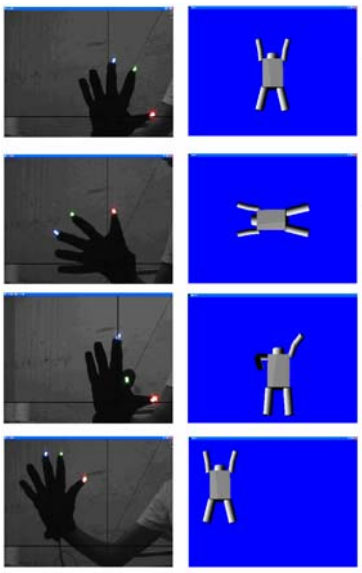


Figure 4. Controlling 3D robot

 ISBN 978-80-86943-99-2

Example 2

The second example demonstrates manipulating 3D object that has foots of a multi joint. Each foot consists of many parts linearly chained. Fig. 5 shows a screen shot during the manipulation. Actor's thumb, forefinger, and middle finger correspond to the three feet of 3D object respectively. When the middle finger is bended, the corresponding foot winds smoothly according to the parameter of each joint.

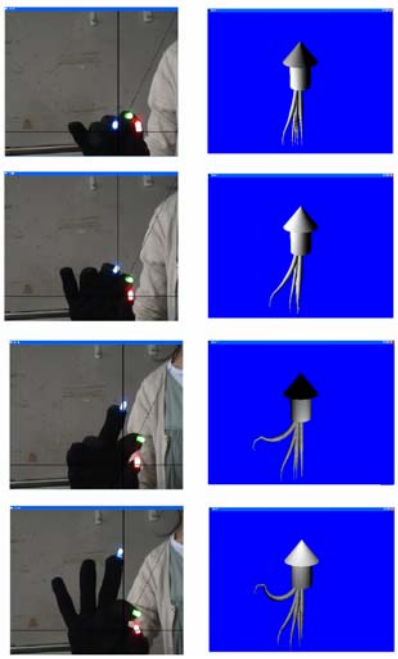


Figure 5.Controlling 3D squid

5. DISCUSSION

A method to control a 3D character by hand is described. In this method, first feature parameters are extracted from motion capture with small number of markers and then, these parameters are transferred as feature parameters to the target character. The experimental use showed that the movement of the whole body of a 3D character can easily be controlled by hand movement as if playing a puppet. Moreover, by spreading a single feature parameter to multiple joints, a character with large-scale set of joints can be controlled. Thus, variation of controllable 3D characters was extended.

Recent researches in motion capture are making efforts to reduce the number of required markers using structural dependency among posture parameters to be captured. Our work, to the contrary, aims to extract as many 'independent' parameters as possible, independent in the sense that each can be independently controlled by the user. In this sense, both motion capture with reduced markers and DPS aim to make maximal use of available information. Note here that the size of parameters extracted by DPS presented in this article is maximum because the degree of freedom of the set of marker is equal to the number of parameters extracted.

These two methods have, however, different intended applications. DPS is suitable for manipulating a character different far from the captured shape, whereas motion capture is for reproducing realistic motion of a character of similar shape.

DPS controls only the movement of 3D character. However, it can also be applied to controlling the speed of the operation or the shape change such as expansion and contraction. Generally, it is applicable to any parametrizable quantity.

Future work

The control of the surface shape by the movement of the hand will be the next step of this research.

ACKNOWLEDGMENTS

The authors thank to Akinori YOSHINO (graduate of Hosei Univ.) for a part of the program used in this system.

REFERENCES

[Att89a] Attwood,C.I., G.D.Sullivan,and K.Baker, Model based recognition of human posture using single synthetic images. *Proc.* the fifth Alvey Vision Conference, pp25-30,1989.

[Cha05a]Chai,J. , and Jessica K. Hodgins, Performance Animation from Low-dimensional Control Signals. *Proc.* ACM SIGGRAPH, No. 3, pp.686-696,2005.

[Kak00a] Kakadiaris,L and D.,Metaxas. Model-based estimation of 3D human motion. IEEE Transactions on Pattern analysis and machine intelligence. Vol.22, No.12,2000.

[Las00a]Laszlo.J, M., van de Panne, and E., Fiume. Interactive Control for Physically-Based Animation. *Proc*. ACM SIGGRAPH, pp. 201-209, 2000

[Nor93a]Norman I. Badler, Michael J. Hollick, John P. Granieri: Real-Time Control of a Virtual Human Using Minimal Sensors. Presence 2(1), pp.82-86, 1993

[Sem98a]Semwal, S., Hightower, R., and Stansfield, S. 1998. Mapping algorithms for real-time control of an avatar using eight sensors. Presence 7, 1, 1-21.

[Ued03a] Ueda,E, Y.Matsumoto, M.Imai, T.Ogasawara. Hand pose estimation for vision-based human interface. IEEE Transactions on Industrial Electronics,Vol.50, No.4,pp676-684,August,2003.

[Yin03a]Yin, K., Pai, D. K, An interactive animation system. *In Proc.* ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp.329-338, 2003.

[Nin]Nintendo,Wii Website http://wiiportal.nintendo-europe.com/13.html

POSTER : A Method for Generating Developments using Decomposition into the Meaningful Components of 3D Polygon Models

Nishio,K.* Iwaizumi, T.** Nakamura, N.* Kobori, K.*

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

**SANYO
1-1
Sanyo-cho, Sanyo-cho, Daito
574-8534, Osaka, JAPAN

nishio@is.oit.a.jp IWAI023108@dt.sanyo.co.jp nakamura@ggl.is.oit.ac.jp kobori@is.oit.ac.jp

ABSTRACT

We propose a method for generating developments from 3D polygon models automatically. The conventional method generates the developments whose components are not interfering each together, by using collision detection between all polygons. However, for the model which consists of a large number of polygons, it is necessary to decompose the development into several parts manually. Therefore it is difficult to generate the development which is easy to be assembled. Our method decomposes the polygon model into meaningful components such as arms, legs, and so on, and develops them. This makes it easy to understand which parts should be glued together, and handcraft bending or folding the developments when a user assembles the paper craft.

Keywords

PAPER CRAFT, DEVELOPMENT, MODELING

1. INTRODUCTION

Recently, CAD or CAM systems are used for designing industrial products. Those make it easy to edit or reuse those designed shapes.

However, those designed shapes are only displayed on the screen as an image, and we cannot touch them with our hand and watch around. Therefore the paper craft has gotten an attention as a easy way to make models recently. With paper craft models, it is useful from the perspective of not only engineering, but also education, to sense those models in three dimensions.

The development of polygon model can be generated by searching all developing patterns. However in the large number of polygons, it takes longer time to be processed. Therefore, some of proposed methods [Mit04]generate developments in consideration of the length of the cutting line, the area of circumscribed rectangle of the development or branch connections of polygons on the development. In addition, in case of large number of polygons, the polygon model is decomposed into several parts not to reduce the number of polygons composing a piece of the development or not to interfere each together.

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

We propose a method which decomposes complex models with large number of polygons into several components and generates developments which is easy to be assembled. Models are divided at the concave and convex region regarding as characteristics and components are developed to be cylindrical.

2. PARTS DECOMPOSING

Our method decomposes models based on Katz's method[Kat03]. Our method can decompose models into meaningful parts in case of animal shape on concave edges.

The process consists of following procedures.

(1) Generate the dual graph representing the connections of polygons.

(2) Calculate and set the costs to the arc of the graph

(3) Select representative nodes in the dual graph

(4) Categorize the nodes as determined region or undetermined region

(5) Set braches of undetermined region to max-flow regarding as a network with max-flow theorem

(6) Divide the model (part) into two parts at the parting line determined with max-flow min-cut theorem

(7) Continue (3)-(6) procedures recursively

(8) Decompose the polygon model with reference to dual graph

2.1 Calculating the costs

While boundary representation model represents vertices, edges and faces, dual graph represents connection of faces. A Dual Graph G* for Graph G is defined as follows. face f in G is related to node f* in G* and edge e in G is related to node e* in G*. Fig.1 shows an example of the relationship between G and G*.

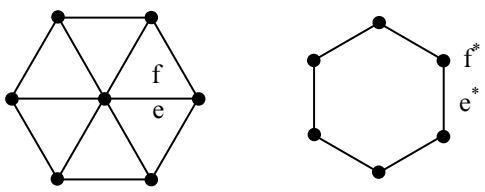


Figure 1. Dual graph

The dual graph related to the graph representing edges and vertices of polygon model, can be considered as a graph which represents connection of faces. Each cost which is set to an arc of dual graph, is calculated with dihedral angle of polygons sharing the edge and geodesic distance between centers of polygons. Eq.(1) shows the cost form the dihedral angle.

$$AngDist(\alpha_{ij}) = \eta(1 - |\cos\alpha_{ij}|) \qquad (1)$$

where α_{ij} is dihedral angle between face f_i and f_j, and η is weighting coefficient.

$GeoDist(f_i, f_j)$ is the geodesic distance between centers of face f_i and f_j as shown in fig.2. Eq.(2) is the cost which is set to the edge.

$$Cost(f_i, f_j) = \delta \frac{GeoDist(f_i, f_j)}{GeoDist_{avg}} + (1-\delta)\frac{AngDist(\alpha_{ij})}{AngDist_{avg}} \qquad (2)$$

where $GeoDist_{avg}$ is average of geodesic distances and $AngDist_{avg}$ is average of dihedral angles.

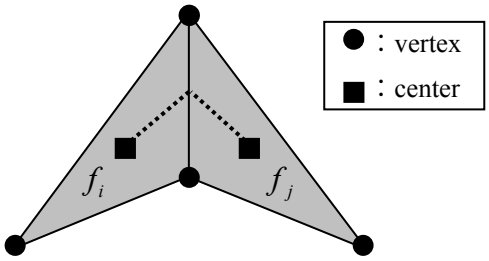


Figure 2. Geodesic distance

2.2 Selecting Representative nodes

This procedure selects a pair of node whose distance is the longest, using Dijkstra's algorithm. We call them representative node. To decompose a polygon model, the process categorizes nodes as determined or undetermined region. The nodes near the representative nodes are categorized as determined region and other nodes are categorized as undetermined region. The nodes categorized as undetermined region can be candidate of cutting path.

2.3 Dividing undetermined region

To divide undetermined region, this procedure regards undetermined region as a network and searches cutting line using max-flow min-cut theorem. The advantage of our method is reduction of meandering of cutting line.

2.4 Termination conditions

The decomposing procedure continues recursively until each decomposed part meets termination conditions as follows.

(a) The number of faces is small enough.

(b) The distance between representative nodes becomes significantly short after division.

(c) The difference between maximum value and minimum value of dihedral angles of a decomposed part.

3. GENERATING DEVELOPMENTS

In conventional method, the process generates developments by cutting out top face and bottom face and by rolling out other faces as shown fig.3.

 ISBN 978-80-86943-99-2

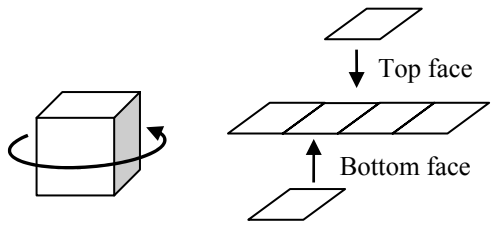


Figure 3. Parts developing

Our method is also based on conventional one. In addition, our method makes it easy to cut out pieces form development and handcraft bending or folding the developments when a user assembles the paper craft. As shown in fig.4, total length of cutting line of left one is longer than right one. Therefore right one is easy to cut out.

In our method, the process modifies developments so that cutting line becomes shorter by attaching and detaching developments each other.

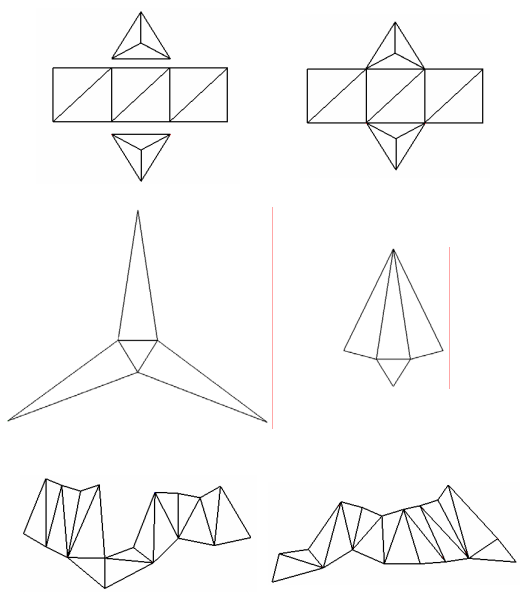


Figure 4. Developments modification

4. RESULTS

To evaluate effectiveness of proposed method, we apply it to develop some three dimensional polygon models as shown in fig.5. Fig.6 shows decomposed models. Fig.7 and 8 shows developments of decomposed parts generated by using our method. In addition, we assembled these developments. Fig.9 shows resulting paper craft models.

Those developments were decomposed into cylindrical parts and rolled out. Therefore it was easy to cut out from development sheet. And also it is easy to bend and grew together.

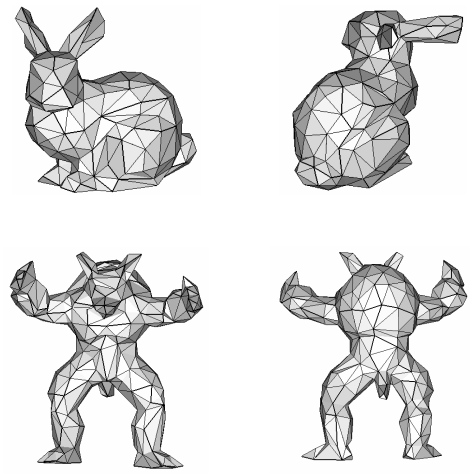


Figure 5. Experimental models

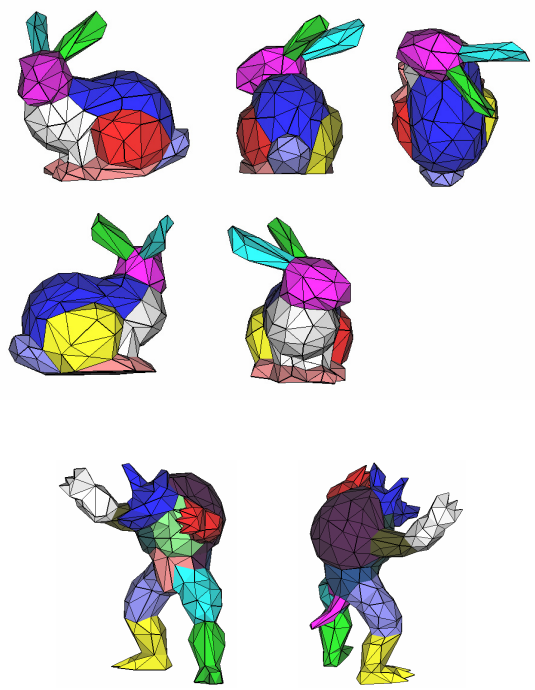


Figure 6. Parts decomposing

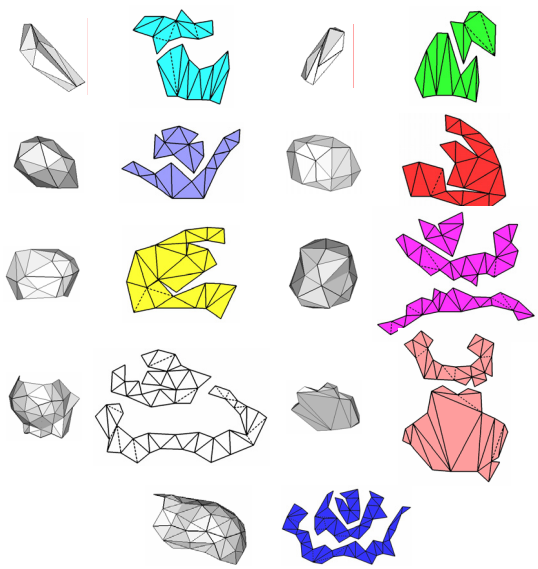


Figure 7. Developments for Bunny

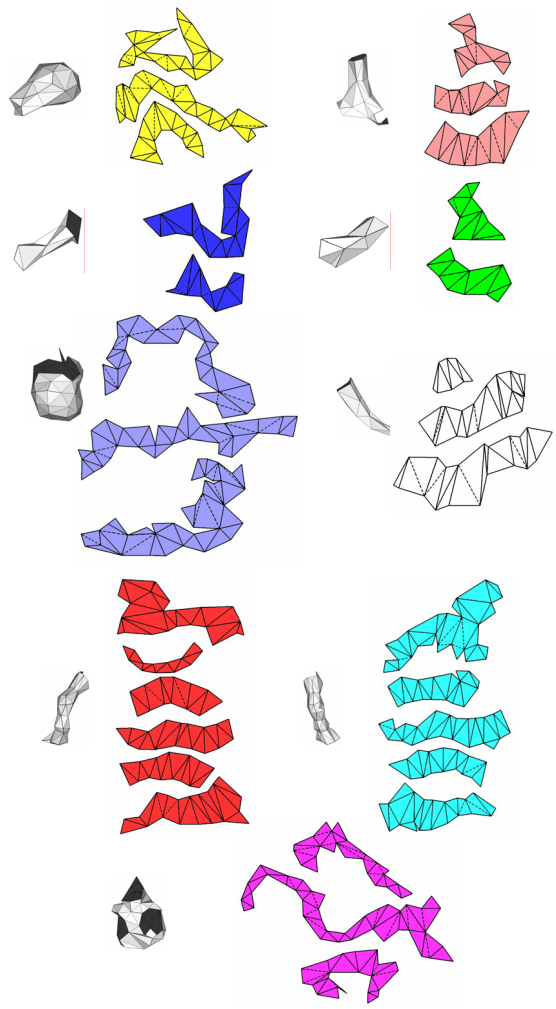


Figure 8. Developments for Armadillo

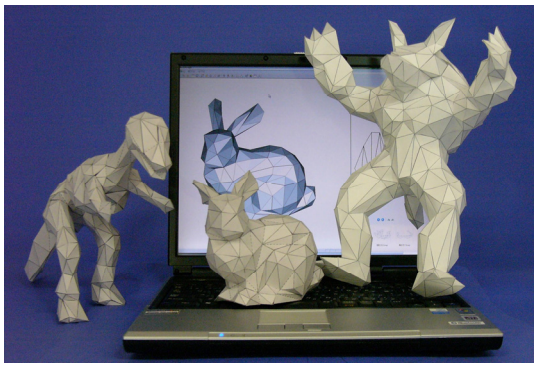


Figure 9. Resulting paper crafts

5. CONCLUSIONS

In this paper, we have described about methods of generating development for polygon model and proposed a method of decomposing polygon model into several parts and generating developments.

The Experimental results shows that our method is useful to generate developments and they are easy to assemble.

6. REFERENCES

[Mit04] Mitani, J. and Suzuki, H., Making Papercraft Toys from Mcshcs using Strip-based Approximate Unfolding, ACM Transactions on Graphics, pp.259-263,Vol.23, No.3, 2004.

[Kat03] Katz, S. and Tal, A. Hierarchical Mesh Decomposition using Fuzzy Clustering and Cuts, ACM Transactions on Graphics, pp.13-18, Vol.22, No.3, 2003.

Customizable Tool for Complex Systems Visualization

Meva Dodo, Patrice Torguet, Michelle Sibilla, Jean Pierre Jessel
Institut de Recherche en Informatique de Toulouse
118 Route de Narbonne
Toulouse, FRANCE
dodo@irit.fr, torguet@irit.fr, sibilla@irit.fr, jessel@irit.fr

ABSTRACT

The last generation of tools used for complex systems monitoring are based on 3D visualization. No doubt, they have allowed to make several improvements but the large amount of dynamical information gathered from heterogeneous environments and the growth of management protocols still make monitoring tasks an interesting research topic. A set of techniques is presented here to manage any complex system by offering a customizable 3D visualization tool associated with a standard information model, which enables to manage environments, devices and applications uniformly regardless of industry and vendor. The aim of this work is to allow tools for any complex system management and monitoring, but we have chosen clusters/grids of networked computers as a first domain of study.

Keywords

3D Visualization, Computer network management, Common Information Model

1. INTRODUCTION

Almost all complex systems are built with heterogeneous environments. The main problem of such systems is to find ways and methods to achieve efficiently the management and monitoring tasks because each environment proposes its own management protocol and thus its own data model. To cope with this problem, the trend is now to work with a standard architecture allowing the integration of all environments to manage and the use of a common data model in order to provide a homogeneous view of management information.

The main issue with visualizing complex systems is how to represent the information in the most effective and aesthetic way in order to easily understand all events and resources and allow the user to control his/her tasks. Therefore, we think that proposing tools, which enable to customize the 3D visualization, will really improve the monitoring and management tasks because it is always easier to understand a system that can be well controlled. Other advantages of customizing the visualization are, firstly to enable monitoring any kind of system because the metaphors are not dedicated to a specific system, and secondly it allows several users to have different representations of the system and events according to their needs.

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

Computer network management and monitoring is a field where several works have been done. Almost all visualization techniques have been used to offer efficient tools in order to facilitate manager's tasks but several efforts should be done in relation to the huge amount of dynamical information and the increasing number of nodes.

To deal with this problem, many traditional tools use 2D visualization techniques [Hew05] [Brad05] [Deb00]. 3D visualization techniques [Ben99] [Brow00] [Bmc05] have facilitated the understanding of a complex system by eliminating some restrictive issues of 2D techniques such as display clutter and device overlaps. The last generation of tools uses virtual reality techniques [Jam98] [Rus00] that facilitate the understanding of complex system, for example, by displaying 3D virtual rooms where many users represented by virtual humans cooperate in monitoring the entire system.

Whatever technique and algorithm are used, the huge amount of dynamical information will always be a major barrier for understanding a large and complex system. Our approach is not simply to allow visualizing an entire system but also only visualizing in 3D all necessary management information for monitoring tasks according to the events, which

occurred. Therefore, the displayed information is only the one that is generated by the interesting events. As our main aim is to give the users the control of the system they monitor, this paper will focus on how to allow managers customizing the visualization according to their needs and their roles.

2. STANDARD INFORMATION MODEL REQUIREMENT

To deal with the variety of technology used in each managed device the DMTF (Distributed Management Task Force) proposes an object-oriented model, CIM (Common Information Model), aiming to provide a common way to represent the computing and networking elements and the relationships between them.

Information collection

Most tools for complex network management use SNMP agents running on each device to collect management information. However, the problem becomes very complex when the system to study is built with different environments each proposing its own management protocol. With the DMTF effort, it becomes possible to manage complex systems built with heterogeneous environments and applications (figure 1).

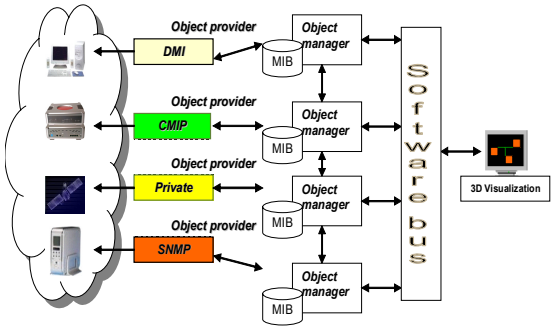


Figure 1: Collecting information process

To cope with such systems, the DMTF proposes the WBEM (Web-Based Enterprise Management) architecture [Wbe05] that introduces the notions of object manager (OM) and object provider (OP). An object provider is a management agent responsible with collecting management information and the integration of all specific environments (for example UNIX, SNMP, LDAP, etc). Once collected, the data is stored in a management information base (MIB) that characterizes an object manager. An object manager is an application responsible of one or several management domains (functional or structural). In such a system, the user sets up the list of interesting information via a query system. The list of queries is sent to the object managers responsible of the domain to study, which requests the appropriate manager agents (object providers). We are developing an interface that will allow user specifying graphically the queries.

Information model

In order to provide a common view of computing network elements and relationships, DMTF proposes an object-oriented model CIM [Cim05] for modeling the management information base. Through the CIM concept, DMTF proposes structural concepts derived from the object paradigm and new specific concepts (qualifier, trigger, indication) expressing constraints and/or meta-information on the structural concepts, and the dynamic of modeled elements.

Using those grammars, a modeling approach with three levels of abstraction is proposed. The “Core Model” introduces high-level abstract classes allowing the organization of the managed elements into physical and logical elements. The “Common Model” is the extension of the “Core Model” to the following domains: system, application, network, physical equipment and functional equipment; and the “Extension Model”: specializes the “Common Model” to the technology domains related to the implementation of the managed resources. Extension Models are defined by individual companies or organizations, but not by the DMTF.

3. VISUALIZATION

Visualization concept

If we want to monitor large-scale systems and particularly when the data exchanges between managed elements is very important for understanding the system, the most efficient way is to use relationship concepts. Indeed, this paradigm facilitates the process of human perception to understand the relational structures of information, thus the system structure. Therefore, our visualization concept consists in displaying in 3D all relevant information and their relationships necessary to understand an event occurring on the system in order to perform necessary actions.

Metaphors

Simplifying the understanding of a complex system first consists in allowing the identification of the main interest of the domain to study. For this, representing management information with metaphors having analogies with the real system largely improves the legibility of information and easily introduces the concept of the system to study. For our project, we have developed several metaphors that users can choose to represent a computer network. Default 3D forms represent devices automatically, such as computer, hub, server, switch, etc. If any of the

simple 3D objects cannot relate visually to the information, it is also possible to add textures, to directly load a VRML object or to use simple 3D objects (cube, cylinder, etc). The size and color of relationships between the elements can change dynamically to indicate the system's behavior, for example bottlenecks, computer failures, etc.

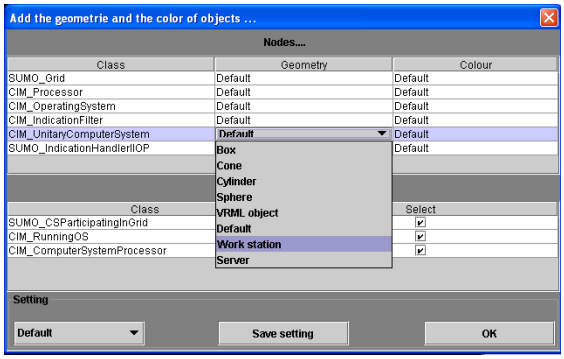


Figure 2: Choice of form and color of 3D objects

Another interesting feature of our tool is the possibility to save a configuration for each user in order to reload it the next time the tool is used (figure 2).

Information mapping

To deal with the huge amount of management information we have chosen the node-link paradigm to represent in 3D management information and their relationships in order to ease the understanding of the phenomenon to study. To avoid the display clutter we display only information about devices or elements concerned by event(s) to which the user subscribed. A partitioning algorithm is used to split the set of information in different groups so that only elements having no links to each other are in the same group. This algorithm has the same principle as the coloration algorithm graph presented in [Wal01].

Our mapping process begins with the choice of metaphors. The user can either customize the 3D representation of each class or let the system generate it automatically (figure 2). For each class, it is possible to choose simple 3D object such as a cube, a cylinder, a sphere or a cone as metaphor. After the choice of geometry, one can color each class so that all its instances are represented by a unique color. If the class colors are kept by default then the application will generate them automatically. One advantage to represent each class by different metaphor is the clarity of the display in order to distinguish easily each group of data. According to the user's needs, it is also possible to change graphically (figure 3) the metaphor of an object or of all the instances of its class.

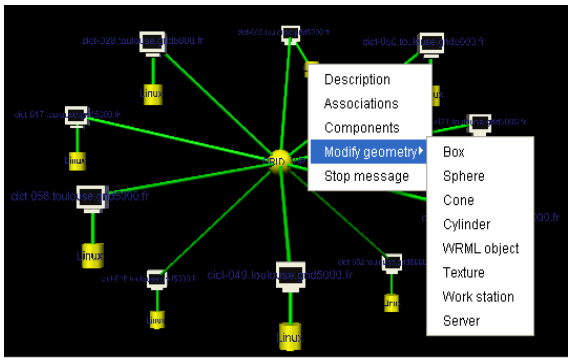


Figure 3: Graphical customizing of geometry of 3D objects

Figure 3 shows a 3D visualization of a cluster where 3D forms represent computer devices and yellow cylinders represent operating systems.

Navigation

Another factor raising the efficiency of visualization is the quality of navigation. Indeed it allows exploring the entire system and getting all details according to the user's needs. We have implemented several techniques to make users explore the system. The first way consists in zooming, in rotating and in translating the entire scene. By always using these mechanisms, it is possible to study independently a specific object. The main advantage of this ability is to rearrange the display in order to get the best presentation. Therefore, if there is a clutter or an overlapping, the user can always enhance the display.

A menu is associated with each object. It enables to get textual information about the clicked object such as its name, original class, its status (for example "down" or "running" for a device) and all other details present in the model. It is also possible to explore the visualization by displaying association or composition relationships between elements for a clicked object.

Event displaying

In order to detect easily all errors occurring in the system or any malfunctions we have implemented two kinds of metaphors. The first metaphor consists in displaying an alert message by modifying the color of the object where the problem is occurring. Therefore, the color of the object changes continuously (figure 4) from the original color to red until a necessary action is carried out.

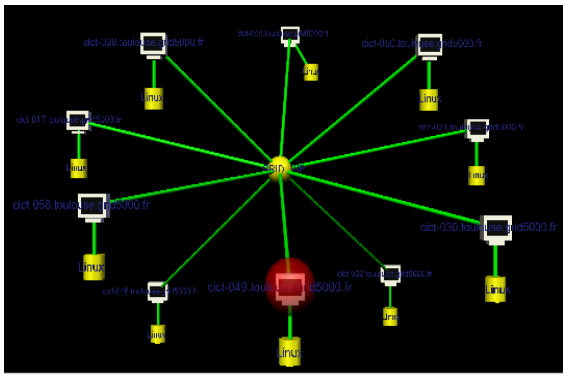


Figure 4: Event display

In the case of objects represented by VRML objects or other textured 3D shapes, a red torus is displayed which is continuously scaled up and down. The appearance of an event changes automatically the position of the source object on the screen in an unoccupied place. The second method is a 3D sound message associated with the color message in order to attract the user's attention and to easily identify the event source.

4. CONCLUSION AND FUTURE WORK

Our work is directed to the problem of management and monitoring of complex systems built with heterogeneous environments. We have found out that the relationship concept is the best way to understand the complex structure of a large system. In our system, each user can have its own configuration of the visualization and save it for the next use of the tool. We have also added a sound message coupled with color or shape changes in order to facilitate the identification of any malfunction in a system.

Our next goal is to integrate dynamic features in the visualization. For this, we will add an interface to allow users to specify the list of interesting events. We are also working on other metaphors and mechanisms in order to enable customizing the appearance of each event to survey.

5. REFERENCES

[Cim05] The Distributed Management Task Force, *The Common Information Model (CIM)*. http://www.dmtf.org/standards/documents/CIM/DSP0111.pdf, Version 2.4, August 2000.

[Ben99] Bénédicte Le Grand, Michel Soto, *Navigation in Huge Information Hierarchies – Application to Network Management*, *ACM NPIVM'99 Workshop,* Kansas City, MI, USA, pp. 56-61, November 2-6, 1999

[Brad05] Bradley Huffaker, *Otter: A general purpose network visualization tool*. Retrieved July 12, 2005, from http://www.caida.org/tools/visualization/otter/paper/

[Bmc05] BMC Software. *The New Vision of Network Management*. Retrieved March 3, 2005, from http://www.atsweb.it/Images/Documenti/BMC_DS_Visualis.pdf

[Brow00] J.A Brown, A.J MCGregor, H-W Braun. "Network Performance Visualization: Insight Through Animation". *Proceedings PAM2000 Passive and Active Measurement Workshop,* Hamilton, New Zealand, pp. 33-41, April 2000

[Deb00] Deborah Estrin, Mark Handley, John Heidemann, Steven McCanne, Ya Xu, *Network Visualization with the VINT Network AniMator*, IEEE Computer, 33 (11), pp.63-68, November 2000

[Hew05] Hewlett-Packard, *Manage dynamic networks with HP OpenView*. Retrieved June 5, 2005, from http://www.managementsoftware.hp.com/solutions/nsm/nsm_bb.pdf

[Jam98] James A. Cubeta, Kirk T. Kern, David D. Egts. VENoM – *Virtual Environment for Network Monitoring*, *Symposium on Advanced Information Processing and Analysis proceedings,* March 1998

[Rus00] Cristina Russo dos Santos, Pascal Gros, Pierre Abel, Didier Loisel, Nicolas Trichaud, Jean-Pierre Paris, *Mapping information onto 3D virtual worlds*
IV'2000, IEEE International Conference on Information Visualization, 19-21 July 2000, London, UK , pp *379 – 386*

[Wal01] Chris Walshaw, *A Multilevel Approach to the Graph Colouring Problem*, *Mathematics Research Report 01/IM/69*, University of Greenwich, London, May, 2001, pp. 153-160

[Wbe05] The Distributed Management Task Force, *The Web-Based Enterprise Management standard (WBEM)*. Retrieved June 5, 2005, from http://www.dmtf.org/standards/wbem

POSTER: Finding Direction of Intersection Curve in Critical Cases of Surface-surface Intersection

Min-jae Oh
Department of Naval Architecture and Ocean Engineering
Seoul National University
Seoul 151-744, KOREA
Mjoh80@snu.ac.kr

Seok Hur
Department of Mathematical Sciences
Seoul National University
Seoul 151-747, KOREA
estone@snu.ac.kr

Tae-wan Kim
Department of Naval Architecture and Ocean Engineering
Seoul National University and RIMSE
Seoul 151-744, KOREA
taewan@snu.ac.kr

ABSTRACT

Determining the topology of intersection curves is one of the important issues of surface-surface intersection problem used in Computer Aided Geometric Design and Computer Graphics. To compute the intersection curves, we first need to determine the topology of the curves. Thomas A. Grandine[Gr97] presented an algorithm to determine topology using partial derivatives of surface intersection equations. When the two surfaces meet tangentially, the differential values of the parameters of the surfaces are not determined in the intersection equations. These cases are called critical cases. In [Ye99] a method of finding the values of the differentials is presented for the case of the contact of order 1. We present general methods for the case of the contact of higher order ≥ 1 using perturbation method. With these results, we can decide starting or ending of the critical boundary point.

Keywords

surface-surface intersection(SSI), topology decision, critical case, tangential intersection, perturbation method.

1. INTRODUCTION

Surface-surface intersection is an important issue, which is used in Computer Aided Design, Computer Aide Geometric Design, Computer Graphics and so on. A topology decision is one part of surface-surface intersection (SSI). To find the intersection curves of two surfaces, first of all, we should figure out the topology of intersection curves of two surfaces. There are many methods are presented in SSI [Ba87, Ba90, Gr97, Gr00, Ho93, Wu99] and, in particular, in finding topology [Gr91, Se88, Se89]. But, in this paper, we focus on the extension of the algorithm presented by T. A. Grandine, which is applied for determining the topology of intersection curves of two surfaces. He presented a method using partial derivatives in surface intersection equations. With Grandine's method, one can determine the topology only in the cases of transversal intersection. So we present an extension of the algorithm for the case of tangential intersection case.

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

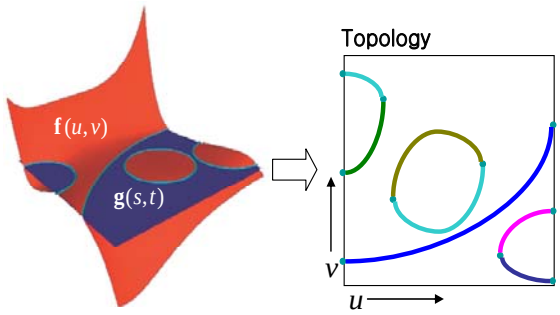


Figure 1. Example of topology

Figure 1 shows an example of topology. There are 7 intersection curves in Figure 1. At each intersection curve, one may determine starting or ending at the boundary and turning points using Grandine's topology decision method. Grandine decides starting or ending by the values of derivatives at each point. But, the intersection is not transversal (that is, two surfaces intersect tangentially) one can not decide the values of derivative with Grandine's method. These cases are called critical cases. In this paper, we consider the critical cases and we present an extension of Grandine's method for such critical cases.

2. PROBLEM DEFINITION

In this paper, we use T. A. Grandine's topology determination method [Ga97] to find topology of SSI. He suggested starting or ending conditions on the boundary points as followings. When two parametric surfaces $\mathbf{f}(u,v)$ and $\mathbf{g}(s,t)$ are given, the next equation would be used for deciding starting or ending of the boundary point $P(s=0)$;

$$\tan^{-1}\frac{\frac{du}{ds}\cos\theta+\frac{dv}{ds}\sin\theta}{\frac{du}{ds}\sin\theta-\frac{dv}{ds}\cos\theta}. \qquad (1)$$

When the value of the equation (1) is positive [resp. negative], the boundary point will be a starting point [resp. ending point]. If the quantity is positive, the intersection curve heads in the parameter space (u,v) of $\mathbf{f}(u,v)$ to the increasing panel direction as s increases.

The values du/ds and dv/ds can be determined by differentiating SSI equation in the case of transversal intersection. The SSI equation writes

$$\mathbf{f}(u,v)-\mathbf{g}(s,t)=0. \qquad (2)$$

Differentiating the equation (2) by the variable s, we get the equation

$$\frac{du}{ds}\mathbf{f}_u+\frac{dv}{ds}\mathbf{f}_v-\mathbf{g}_s-\frac{dt}{ds}\mathbf{g}_t=\mathbf{0}. \qquad (3)$$

The equation (3) can be represented 3×3 linear system of equations

$$\begin{pmatrix} f_u^x & f_v^x & -g_t^x \\ f_u^y & f_v^y & -g_t^y \\ f_u^z & f_v^z & -g_t^z \end{pmatrix}\begin{pmatrix} \frac{du}{ds} \\ \frac{dv}{ds} \\ \frac{dt}{ds} \end{pmatrix}=\begin{pmatrix} g_s^x \\ g_s^y \\ g_s^z \end{pmatrix}. \qquad (4)$$

When the 3×3 matrix is invertible (that is, if two surfaces intersect transversally), we can find the unique vector value $(du/ds, dv/ds, dt/ds)^T$. Then we can decide whether the boundary point $P(s=0)$ is a starting point or an ending point by the equation (1). But, when 3×3 matrix is not invertible (that is, if two surfaces intersect tangentially), we can not determine the topology with the Granidin's method. These cases are called critical cases. We consider such critical cases and suggest a general method of determining the vector value $(du/ds, dv/ds, dt/ds)^T$.

When two surfaces meet tangentially at boundary $P(s=0)$, we cannot determine the values du/ds and dv/ds with the equation (4). For the simple tangential intersection, that is, for the contact of order one, Ye-Ma[Ye99] presented a method of finding the tangent vector of the intersection curve. They used the fact that the two surfaces have the same normal curvatures on the intersection curve. If we know the tangent vector of the intersection curve we can find the vector $(du/ds, dv/ds, dt/ds)^T$ [Oh06]. But the intersection is contact order at least 2 this method is not applicable.

3. TOPOLOGY RESOLUTION OF CRITICAL CASES

Let S_1 and S_2 be two analytic surfaces in R^3 defined by the following parametric representations.

$$\begin{aligned} S_1 &= \{\mathbf{f}(u,v)=(f^x(u,v), f^y(u,v), f^z(u,v)) : 0\le u,v\le 1\} \\ S_2 &= \{\mathbf{g}(s,t)=(g^x(s,t), g^y(s,t), g^z(s,t)) : 0\le s,t\le 1\}. \end{aligned} \qquad (5)$$

We assume that the two surfaces intersect tangentially on an intersection curve l. We want to find the vector value $(du/ds, dv/ds, dt/ds)^T$ of the intersection curve on the boundary $s=0$. If we assume that the intersection curve l is parameterized locally in a small neighborhood of $P(s=0)$ by the variable s, the variables u, v and t can be represented analytically by the variable s in small neighborhood of $P(s=0)$. Since two surfaces have the same tangent plane on l, there is an invertible matrix $A(s)$ such that

$$(\mathbf{f}_u(s) \quad \mathbf{f}_v(s))A(s)=(\mathbf{g}_s(s) \quad \mathbf{g}_t(s)). \qquad (6)$$

And, on l, the fact $\partial/\partial s(\mathbf{f}(u,v)-\mathbf{g}(s,t))=0$ implies the equation (3).

Let $A(0)=\begin{pmatrix} \alpha & \gamma \\ \beta & \delta \end{pmatrix}$ be the invertible matrix at $s=0$. According to the relation (6), we can write

$$\frac{du}{ds}\mathbf{f}_u+\frac{dv}{ds}\mathbf{f}_v=\frac{dt}{ds}\gamma\mathbf{f}_u+\frac{dt}{ds}\delta\mathbf{f}_v+\alpha\mathbf{f}_u+\beta\mathbf{f}_v. \qquad (7)$$

This equation implies the relation

$$\begin{pmatrix} \frac{du}{ds} \\ \frac{dv}{ds} \end{pmatrix}=\begin{pmatrix} \alpha & \gamma \\ \beta & \delta \end{pmatrix}\begin{pmatrix} \frac{ds}{ds} \\ \frac{dt}{ds} \end{pmatrix}=\begin{pmatrix} \alpha & \gamma \\ \beta & \delta \end{pmatrix}\begin{pmatrix} 1 \\ \frac{dt}{ds} \end{pmatrix}. \qquad (8)$$

Now we have the following lemma,

Lemma If two surfaces intersect tangentially at the boundary point $P(s=0)$, finding the vector value $(du/ds, dv/ds, dt/ds)^T$ is equivalent to finding the value dt/ds.

Now we perturb analytically one of the surface equations, for instance, $S_2(\varepsilon)=\{\mathbf{g}(s,t,\varepsilon):0\le s,t\le 1\}$ so that the intersection is no longer tangential for sufficiently small $\varepsilon\neq 0$. We remark that the Sard's theorem [Sp99] guarantees the existence of such perturbations and that the method of such perturbations will be various. Analytic perturbation means the vector valued function $\mathbf{g}(s,t,\varepsilon)$ is an analytic function with respect to ε and $\mathbf{g}(s,t,0)=\mathbf{g}(s,t)$. After such perturbation the equation (4) can be written

$$\begin{pmatrix}\mathbf{f}_u & \mathbf{f}_v & -\mathbf{g}_t(\varepsilon)\end{pmatrix}\begin{pmatrix}\frac{du}{ds}(\varepsilon) & \frac{dv}{ds}(\varepsilon) & \frac{dt}{ds}(\varepsilon)\end{pmatrix}^T=\mathbf{g}_s(\varepsilon). \quad (9)$$

Since the perturbation is analytic with respect to ε the values $\mathbf{g}_t(\varepsilon), \begin{pmatrix}\frac{du}{ds}(\varepsilon) & \frac{dv}{ds}(\varepsilon) & \frac{dt}{ds}(\varepsilon)\end{pmatrix}^T$ and $\mathbf{g}_s(\varepsilon)$ are analytic with respect to ε for sufficiently small $\varepsilon\neq 0$. These facts write

$$\mathbf{g}_s(\varepsilon)=\mathbf{g}_s(0)+\sum_{k\ge 1}\varepsilon^k\mathbf{g}_s^k=\alpha\mathbf{f}_u+\beta\mathbf{f}_v+\sum_{k\ge 1}\varepsilon^k\mathbf{g}_s^k,$$

$$\mathbf{g}_t(\varepsilon)=\mathbf{g}_t(0)+\sum_{k\ge 1}\varepsilon^k\mathbf{g}_t^k=\gamma\mathbf{f}_u+\delta\mathbf{f}_v+\sum_{k\ge 1}\varepsilon^k\mathbf{g}_t^k,$$

$$\begin{aligned}\frac{du}{ds}(\varepsilon)&=\frac{du}{ds}(0)+\sum_{k\ge 1}\varepsilon^k\left(\frac{du}{ds}\right)_k\\&=\alpha+\gamma\frac{dt}{ds}(0)+\sum_{k\ge 1}\varepsilon^k\left(\frac{du}{ds}\right)_k,\end{aligned} \quad (10)$$

$$\begin{aligned}\frac{dv}{ds}(\varepsilon)&=\frac{dv}{ds}(0)+\sum_{k\ge 1}\varepsilon^k\left(\frac{dv}{ds}\right)_k\\&=\beta+\delta\frac{dt}{ds}(0)+\sum_{k\ge 1}\varepsilon^k\left(\frac{dv}{ds}\right)_k,\end{aligned}$$

$$\frac{dt}{ds}(\varepsilon)=\frac{dt}{ds}(0)+\sum_{k\ge 1}\varepsilon^k\left(\frac{dt}{ds}\right)_{k1}.$$

Then the equations (9) and (10) imply

$$\begin{aligned}&\mathbf{f}_u\left[\alpha+\gamma\frac{dt}{ds}(0)+\sum_{k\ge 1}\varepsilon^k\left(\frac{du}{ds}\right)_k\right]\\&+\mathbf{f}_v\left[\frac{dv}{ds}(0)\beta+\delta\frac{dt}{ds}(0)+\sum_{k\ge 1}\varepsilon^k\left(\frac{dv}{ds}\right)_k\right]\\&-\left[\gamma\mathbf{f}_u+\delta\mathbf{f}_v+\sum_{k\ge 1}\varepsilon^k\mathbf{g}_t^k\right]\left[\frac{dt}{ds}(0)+\sum_{k\ge 1}\varepsilon^k\left(\frac{dt}{ds}\right)_k\right]\\&-\left[\alpha\mathbf{f}_u+\beta\mathbf{f}_v+\sum_{k\ge 1}\varepsilon^k\mathbf{g}_s^k\right]=0.\end{aligned}$$

Consequently, we have an expansion with respect to ε

$$\begin{aligned}&\mathbf{f}_u\left[\alpha+\gamma\frac{dt}{ds}(0)\right]+\mathbf{f}_v\left[\beta+\delta\frac{dt}{ds}(0)\right]\\&-(\gamma\mathbf{f}_u+\delta\mathbf{f}_v)\frac{dt}{ds}(0)-(\alpha\mathbf{f}_u+\beta\mathbf{f}_v)\\&+\varepsilon\begin{bmatrix}\mathbf{f}_u\left(\frac{du}{ds}\right)_1+\mathbf{f}_v\left(\frac{dv}{ds}\right)_1\\-(\gamma\mathbf{f}_u+\delta\mathbf{f}_v)\left(\frac{dt}{ds}\right)_1-\mathbf{g}_t^1\frac{dt}{ds}(0)-\mathbf{g}_s^1\end{bmatrix}\\&+\sum_{k\ge 2}\varepsilon^kR_k=0.\end{aligned} \quad (11)$$

This equality implies the vanishing conditions $R_k\equiv 0$, for all $k\ge 1$.

The vanishing of constant term is trivial. And the vanishing of the coefficient of ε gives

$$\begin{aligned}&\mathbf{f}_u\left(\frac{du}{ds}\right)_1+\mathbf{f}_v\left(\frac{dv}{ds}\right)_1-(\gamma\mathbf{f}_u+\delta\mathbf{f}_v)\left(\frac{dt}{ds}\right)_1-\mathbf{g}_s^1\\&=\mathbf{g}_t^1\frac{dt}{ds}(0).\end{aligned} \quad (12)$$

If $\varepsilon\neq 0$, two surfaces sustains transversal intersection, the values $(du/ds)_1,(dv/ds)_1$ and $(dt/ds)_1$ can be evaluated by the equation (4). The values $\mathbf{g}_s^k$ and $\mathbf{g}_t^k$ can be calculated directly by the assumption of analyticity of the perturbation. So we can find the value $dt/ds(\varepsilon=0)$ in above equation when the vector $\mathbf{g}_t^1$ does not vanish. If the value $\mathbf{g}_t^1=0$ we can not find exact value of $dt/ds(\varepsilon=0)$ in the equation (12) and there are two situations depending on the value of the left side of the equality (12). As the first case, left side does not vanish, and then, consequently, there is no such value $dt/ds(\varepsilon=0)$ satisfying the equality. This means the point $P(s=0)$ is an isolate tangential intersection point or the direction of intersection is t-axis, that is, $dt/ds=\infty$. Secondly, the left side vanishes, and then the equality (12) satisfies for any value of $dt/ds(\varepsilon=0)$. So we must use the next vanishing condition $R_2\equiv 0$. If we can not decide the value $dt/ds(\varepsilon=0)$ with the condition $R_2\equiv 0$ we continue this procedure till we could find the value $dt/ds(\varepsilon=0)$. We remark the functions R_k are polynomials with respect to the parameters. Since the ring of polynomials is Noetherian, the ideal generated by R_k has finite generators. Vanishing of these finite generators gives the value $dt/ds(\varepsilon=0)$.

We remark that the difficulty of calculation depends on the method of perturbations. In practical calculation, we can use the following simplified form $dt/ds(\varepsilon=0)=\lim_{\varepsilon\to 0} dt/ds(\varepsilon)$.

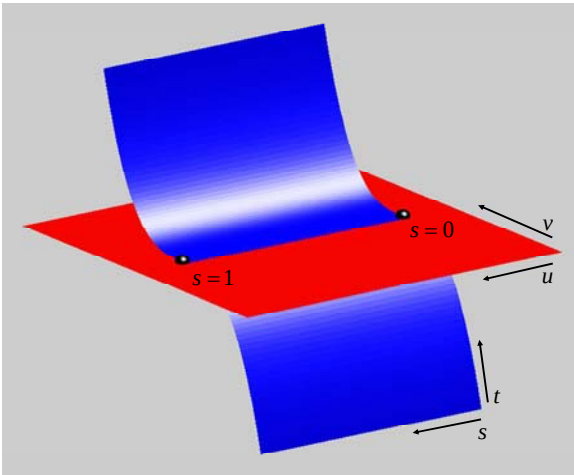


Figure 2. The critical boundary points of contact of order two

As an example (see Figure2), we consider a perturbation

$$S_1=\{\mathbf{f}(u,v)=(u,v,0):0\le u,v\le 1\}$$
$$S_2=\{\mathbf{g}(s,t)=(0.7s+0.15,0.7t+0.15,(t-0.5)^3+\varepsilon):0\le s,t\le 1\}.$$

The vanishing condition gives $dt/ds\cdot(\varepsilon)^{\frac{2}{3}}=0$. This means $dt/ds=0$ for $\varepsilon\neq 0$ and consequently we have $dt/ds(\varepsilon=0)=\lim_{\varepsilon\to 0} dt/ds(\varepsilon)=0$.

4. FUTURE WORKS

We have considered critical case of boundary points. Another critical case is concerning the turning point in the Grandine's method. For example two pipes intersect with cross intersection curves. In such case we cannot decide starting or ending of the cross turning point with the Grandine's method.

ACKNOWLEDGEMENTS

This work was supported by grant No. R01-2005-000-11257-0 from the Basic Research Program of the KOSEF and Seoul Digital Center for Research and Development of Ubiquitous Computing Technologies from Seoul City.

REFERENCES

[Ba87] G. Farin, M. J. Barnhill, R.E. and B.R. Piper. Surface/surface intersection, Computer Aided Geometric Design, 4(1-2):3-16, 1987.

[Ba90] R. E. Barnhill and S. N. Kersey, A marching method for parametric surface/surface intersection, Computer Aided Geometric Design, 7(1-4):257-280, 1990.

[Gr91] G. Mullenheim, On determining start points for a surface/surface intersection algorithm, Computer Aided Geometric Design, 8(5):401-408, 1991.

[Gr97] T. A. Grandine and W. K. Frederick, A new approach to the surface intersection problem, Computer Aided Geometric Design, 14(2):111-134, 1997.

[Gr00] T. A. Grandine. Application of contouring, SIAM Review, 42(2):297-315, 2000.

[Ho93] J. Hoschek and D. Lasser. Fundamentals of Computer Aided Geometric Design, A K Peters, 1993.

[Oh06] M.-J. Oh, S. Hur and T.-W. Kim, Topology determination of critical cases in surface-surface intersection, Proceedings of the Korean Society for Industrial and Applied Mathematics, 1(2):149-151, 2006.

[Se88] T. W. Sederberg and R.J. Meyers, Loop detection in surface patch intersections, Computer Aided Geometric Design, 5(2):161-171, 1988.

[Se89] H. N. Christiansen Sederberg, T.W. and S. Katz, An improved test for closed loops in surface intersection, Computer-Aided Design, 21(8):505-508, 1989.

[Sp99] M. Spivak, A Comprehensive Introduction to Differential Geometry, Vol. I, Third Edition, Publish of Perish, Inc. Houston, Texas, 1999.

[Wu99] S.-T. Wu and L. N. Andrade, Marching along a regular surface/surface intersection with circular steps, Computer Aided Geometric Design, 16(4):249-268, 1999.

[Ye99]X. Ye, T. Maekawa, Differential geometry of intersection curves of two surfaces, Computer Aided Geometric Design, 16(8): 767-788, 1999 .

Poster: Experiments with Volumetric Data Interpolation

Alexej Kolcun
Dep. of Appl. Math&Inf.
Institute of Geonics CAS, Ostrava
Czech Republic
kolcun@ugn.cas.cz

Daniel Pravda
Faculty of Science
University of Ostrava
Czech Republic
R02299@student.osu.cz

ABSTRACT

Marching Cubes and Marching Tetrahedra methods expect data in the nodes of used grid. However, very often we need the isosurface extraction from data defined in the cells of the grid. In this paper we compare the criteria for interpolation values from cells to vertices of the structured grid.

Keywords

Volumetric data, interpolation, weighted mean, structured mesh.

1. INTRODUCTION

Marching Cubes [Lore87] (Marching Tetrahedra [Payn90]) algorithms are powerful tools for extracting isosurfaces from scalar volumetric data. These algorithms expect data located at *vertices* of used mesh. In [Pate05] the sensitivity of resulting interpolation on used mesh is analyzed. However, very often we need the isosurface extraction for data defined in the *cells* of the mesh. There are two possibilities how to use "marching" algorithms in this case:

a) We preserve the analyzed values and we make a mesh with new geometry. In the simplest case a dual mesh is constructed (nodes of dual mesh corresponds to cells of original one; nodes of dual mesh are neighbor if cells of original one have common edge).

b) We preserve the geometry of the mesh and the values from cells to vertices are interpolated.

In the paper the second approach is used. We shall use next abbreviation:

$C = \{c_i, i = 1, \cdots n_C\}$ – conform set of polygonal cells in 2D (polyhedral in 3D),

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

$|C| = n_C$ – cardinality of the set C, $\|c_i\|$ – volume of the element c_i,

$V = \{v_i, i = 1, \cdots n_V\}$ – set of the mesh vertices,

C_v – set of the cells, which are incident to vertex *v*.

Four different weighted averages are analyzed for simple test tasks. Structured meshes are used.

2. INTERPOLATION STRATEGIES

Let the values of the scalar field F are known for the cells of the mesh. We can calculate the value of this field at the vertex v as the weighted arithmetic mean of the values in incident cells:

$$F^{calc}(v) = \sum_{c \in C_v} a_c F(c).$$

We shell analyze four methods for weights computation.

Simple Arithmetic Mean $a_c = 1/|C_v|$.

Due to simplicity of computation this strategy is used very often.

Volume Weighted Mean $a_c = \|c\| \Big/ \sum_{d \in C_v} \|d\|$.

Idea of the volume averaging is that the influence of coincident cells at the common vertex is proportional to their size.

Volume Reciprocal Mean $a_c = \frac{1}{\|c\|} \Big/ \sum_{d \in C_v} \frac{1}{\|d\|}$

This strategy seems to be in contradiction with previous one. However, it can be proved that for

1. linear functions $F(x_1, \cdots x_n) = r_0 + \sum_{i=1}^{n} r_i x_i$,

2. orthogonal meshes, with
   $V = \{x_{1,1}, \cdots x_{1,m_1}\} \times \cdots \times \{x_{n,1}, \cdots x_{n,m_n}\}$

3. the value of the cell $F(c) = F(\bar{v})$, where $\bar{v}$ is the center of the cell c ,

this strategy gives the exact value: $F^{calc}(v) = F(v)$.

Angle Averaging $a_c = \alpha_{v,c} / q$.

Here $\alpha_{v,c}$ is the angle at the vertex v within the cell c . $q = 2\pi$ in 2D and $q = 4\pi$ in 3D.

3. TWO-DIMENSIONAL MESH TESTS

We compare the strategies mentioned above for two-dimensional grids. The goal is to gain a basic idea of behavior of the interpolation strategies while changing the shape and size of cells surrounding the given vertex.

Two types of 3x3 grid with 4 rectangular cells (Fig. 1a, 1b) are used. We move the central vertex of the grid and thus we gain grids with different geometries. Different paths for moving central vertex are defined: 1–3 for regular grid (Fig. 1a)), 4–5 for orthogonal one (Fig.1b)). Each starts at the position, when the grid is rectilinear, and passes along the line. In the Fig.1c) one of the grids for the path 2 is shown.

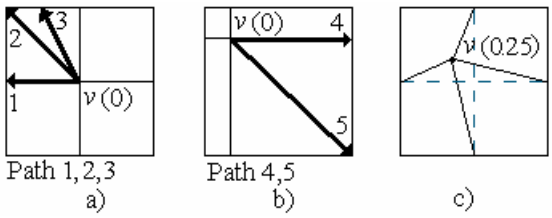


Figure 1.

For each position of the central vertex *v*, the value of each cell is calculated as

$$F(c) = \frac{\iint_c F(x,y)dxdy}{\|c\|} .$$

Fig. 2 shows the values of the relative difference

$$\frac{F^{calc}(v(t)) - F(v(t))}{F(v(t))} \times 100$$

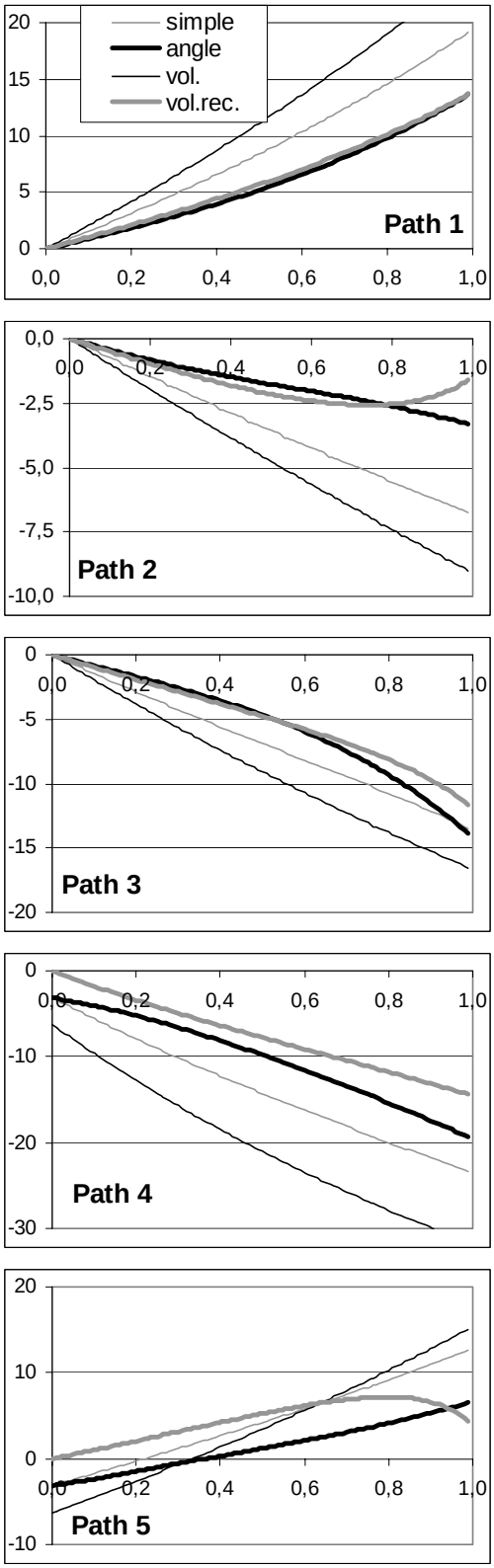


Figure 2.

at the vertex $v(t)$ for the liner function $F(x,y) = 2x + 3y$. The argument of the horizontal

 ISBN 978-80-86943-99-2

axis $t \in \langle 0,1 \rangle$ denotes the position of the vertex $v(t)$ according to the Fig. 1.

We can see that the strategies of volume reciprocal mean and angle averaging give similar results and are significantly better then simple mean. The volume averaging strategy produces the worst results.

4. TESTS IN 3D

We use two three-dimensional structured grids for a comparison of the interpolation strategies.

First one is based on regular rectangular grid with 11x11x11 vertices. Coordinates of inner vertices are randomly shifted, 20 % of the length of the edge at most.

Second one is based on rectangular grid with 8x8x8 vertices refined near one corner – Fig.3. Coordinates of inner vertices are randomly shifted, 5% of the length of the edge at most.

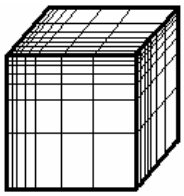


Figure 3.

Tests are made for hexahedral meshes and also for tetrahedral ones. We obtain tetrahedral mesh dividing each hexahedral cell according to the scheme in the Fig. 4.

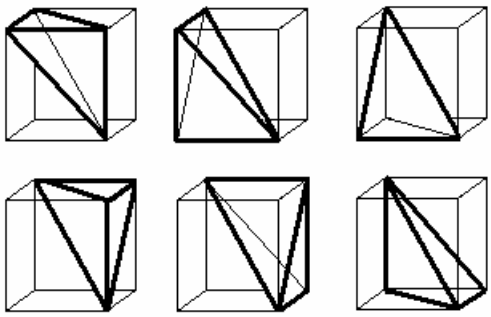


Figure 4.

Linear function $F(x, y, z) = 2x + 3y + 4z$ is used.

Values of the global difference

$$D = \sum_{v \in V} \left(F^{calc}(v) - F(v) \right)^2$$

are in the Table 1 for analyzed strategies.

	Regular grid		Refined grid	
	Hex.	Tetra.	Hex.	Tetra.
Simple mean	215,0	163,3	516,9	358,1
Vol. Weighted	291,7	294,4	2041,8	947,1
Vol. reciprocal	176,1	88,1	5,1	91,6
Angle	33,9	39,8	690,5	157,7

Table 1

5. CONCLUSION

Tests show that the strategy of volume reciprocal averaging, which gives exact solution for rectangular meshes, gives good results in more general cases too (Tab.1 – Refined grid).

For more deformed meshes the strategy of angle averaging gives better results in 3D case (Tab.1 – Regular grid). However, its computational complexity is much larger.

Important result of tests is that the strategy of volume weighted mean appears as improper.

In the future we would like to compare the quality of isosurfaces made by the Marching Cube and by the Marching Tetrahedra algorithms on interpolated data.

ACKNOWLEDGEMENTS

Work is supported by the project of National Research Program of the Czech Republic, project number 1ET400300415.

REFERENCES

[Lore87] Lorensen, W.E., Cline, H.E.:Marching Cubes: A High Resolution 3D Surface Construction Algorithm. Computer Graphics, Vol.21(4), 1987. pp.163-169.

[Pate04] Patera, J.: Iso-Surface Extraction and Approximation Error. Tech.Report DCSE/TR-2004-15. Univ.of West Bohemia, Plzeň 2004.

[Pate05] Patera, J., Skala, V.: Centered Cubic Lattice Method Comparision. In: Proc. of Conf. ALGORIT-MY 2005. STU Bratislava 2005. pp.270-279.

[Payn90] Payne, B.A., Toga, A.V.: Surface Mapping Brain Function on 3D Models. IEEE CG&A 10(5). 1990. pp. 33-41.