The 14-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2006

in co-operation with

EUROGRAPHICS

W S C G ' 2006

University of West Bohemia
Plzen
Czech Republic

January 31 – February 2, 2006

Posters proceedings

Co-Chairs

Joaquim Jorge, Technical University of Lisboa,Lisboa, Portugal
Vaclav Skala, University of West Bohemia, Plzen, Czech Republic

Edited by
Joaquim Jorge, Vaclav Skala

WSCG'2006 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 80-86943-04-6*

Contents

Additional files are available on CD ROM or at http://wscg.zcu.cz

WSCG 2006

International Programme Committee

Bartz,D. (Germany)
Bekaert,P. (Belgium)
Benes,B. (United States)
Bengtsson,E. (Sweden)
Bouatouch,K. (France)
Brunnet,G. (Germany)
Chen,M. (United Kingdom)
Chrysanthou,Y. (Cyprus)
Cohen-Or,D. (Israel)
Coquillart,S. (France)
Debelov,V. (Russia)
Deussen,O. (Germany)
du Buf,H. (Portugal)
Ertl,T. (Germany)
Ferguson,S. (United Kingdom)
Groeller,E. (Austria)
Hauser,H. (Austria)
Hege,H. (Germany)
Jansen,F. (Netherlands)
Jorge,J. (Portugal)
Kalra,P. (India)
Klein,R. (Germany)
Klosowski,J. (United States)
Kobbelt,L. (Germany)
Kruijff,E. (Germany)
Lars,K. (Sweden)
Magnor,M. (Germany)
Moccozet,L. (Switzerland)
Mudur,S. (Canada)
Mueller,K. (United States)
Muller,H. (Germany)
Myszkowski,K. (Germany)
OSullivan,C. (Ireland)
Pasko,A. (Japan)
Peroche,B. (France)
Post,F. (Netherlands)
Puppo,E. (Italy)
Purgathofer,W. (Austria)
Rauterberg,M. (Netherlands)
Rheingans,P. (United States)
Rokita,P. (Poland)
Rossignac,J. (United States)
Rudomin,I. (Mexico)
Sakas,G. (Germany)
Sbert,M. (Spain)
Schaller,N. (United States)
Schilling,A. (Germany)
Schneider,B. (United States)
Schumann,H. (Germany)
Shamir,A. (Israel)
Slusallek,P. (Germany)
Sochor,J. (Czech Republic)
Sumanta,P. (United States)
Szirmay-Kalos,L. (Hungary)
Taubin,G. (United States)
Teschner,M. (Germany)
Velho,L. (Brazil)
Veltkamp,R. (Netherlands)
Weiskopf,D. (Canada)
Westermann,R. (Germany)
Wu,S. (Brazil)
Wuethrich,C. (Germany)
Yamaguchi,F. (Japan)
Zara,J. (Czech Republic)
Zemcik,P. (Czech Republic)

WSCG 2006 Board of Reviewers

Adamo-Villani,N. (United States)
Adzhiev,V. (United Kingdom)
Ammon,L. (Switzerland)
Andreadis,I. (Greece)
Aran,M. (Turkey)
Araujo,B. (Portugal)
Aspragathos,N. (Greece)
Bartz,D. (Germany)
Batagelo,H. (Brazil)
Battiato,S. (Italy)
Bekaert,P. (Belgium)
Benes,B. (United States)
Bengtsson,E. (Sweden)
Beyer,J. (Austria)
Biber,P. (Germany)
Bieri,H. (Switzerland)
Bilbao,J. (Spain)
Biri,V. (France)
Bischoff,S. (Germany)
Borchani,M. (France)
Bottino,A. (Italy)
Bouatouch,K. (France)
Brodlie,K. (United Kingdom)
Brunnet,G. (Germany)
Buehler,K. (Austria)
Cohen-Or,D. (Israel)
Coleman,S. (United Kingdom)
Coquillart,S. (France)
Daniel,M. (France)
Danovaro,E. (Italy)
de Aguiar,E. (Germany)
De Decker,B. (Belgium)
Debelov,V. (Russia)
Del Rio,A. (Germany)
Deussen,O. (Germany)
Di Fiore,F. (Belgium)
Diaz,M. (Mexico)
du Buf,H. (Portugal)
Duce,D. (United Kingdom)
Erbacher,R. (United States)
Ertl,T. (Germany)
Feito,F. (Spain)
Felkel,P. (Czech Republic)
Ferguson,S. (United Kingdom)
Fernandes,A. (Portugal)
Flaquer,J. (Spain)
Fleck,S. (Germany)
Francken,Y. (Belgium)
Gagalowicz,A. (France)
Galo,M. (Brazil)
Geraud,T. (France)
Giachetti,A. (Italy)
Giegel,J. (United States)
Groeller,E. (Austria)
Gudukbay,U. (Turkey)
Guerreiro,T. (Portugal)
Habbecke,M. (Germany)
Haber,T. (Belgium)
Hanak,I. (Czech Republic)
Hast,A. (Sweden)
Hauser,H. (Austria)
Havran,V. (Germany)
Hege,H. (Germany)
Hernandez,B. (Mexico)
Herzog,R. (Germany)
Hirschbach,H. (Germany)
Hornung,A. (Germany)
Chen,M. (United Kingdom)
Chrysanthou,Y. (Cyprus)
Isgro,F. (Italy)
Jaillet,F. (France)
Janda,M. (Czech Republic)
Jansen,F. (Netherlands)
Jeschke,S. (Austria)
Jorge,J. (Portugal)
Jota,R. (Portugal)
Kalra,P. (India)
Kavan,L. (Czech Republic)
Keller,K. (United States)
Kipfer,P. (Germany)
Klein,K. (Germany)
Klosowski,J. (United States)
Kobbelt,L. (Germany)
Kolcun,A. (Czech Republic)
Krüger,J. (Germany)
Kruijff,E. (Germany)
Lanquetin,S. (France)
Lars,K. (Sweden)
Leon,A. (Spain)
Leopoldseder,S. (Austria)
Levy,B. (France)
Lintu,A. (Germany)
Linz,C. (Germany)
Lipus,B. (Slovenia)
Magalhaes,L. (Brazil)
Magillo,P. (Italy)
Magnor,M. (Germany)
Mantler,S. (Austria)
McMenemy,K. (United Kingdom)
Millan,E. (Mexico)
Moccozet,L. (Switzerland)
Moltedo,L. (Italy)
Montrucchio,B. (Italy)
Mudur,S. (Canada)
Mueller,K. (United States)
Muller,H. (Germany)
Multon,F. (France)
Myszkowski,K. (Germany)
Nielsen,F. (Japan)
Novotny,M. (Austria)
O'Sullivan,C. (Ireland)
Pasko,A. (Japan)
Patel,D. (Austria)
Patera,J. (Czech Republic)
Pedrini,H. (Brazil)
Perales,F. (Spain)
Peroche,B. (France)
Platis,N. (Greece)
Plemenos,D. (France)
Porcu,M. (Italy)
Post,F. (Netherlands)
Pratikakis,I. (Greece)
Prikryl,J. (Czech Republic)
Puig,A. (Spain)
Puppo,E. (Italy)
Purgathofer,W. (Austria)
Rauterberg,M. (Netherlands)
Reaz,M. (Malaysia)
Reina,G. (Germany)
Renaud,C. (France)
Revelles,J. (Spain)
Ribelles,J. (Spain)
Rodeiro,J. (Spain)
Rojas-Sola,J. (Spain)
Rokita,P. (Poland)

Rose,D. (Germany)
Rossignac,J. (United States)
Rudomin,I. (Mexico)
Sakas,G. (Germany)
Sanna,A. (Italy)
Sbert,M. (Spain)
Scateni,R. (Italy)
Segura,R. (Spain)
Shah,M. (United States)
Shamir,A. (Israel)
Schafhitzel,T. (Germany)
Schaller,N. (United States)
Scherzer,D. (Austria)
Schilling,A. (Germany)
Schneider,B. (United States)
Schneider,J. (Germany)
Scholz,V. (Germany)
Schumann,H. (Germany)
Sips,M. (Germany)
Sitte,R. (Australia)
Slusallek,P. (Germany)
Snoeyink,J. (United States)
Snoeyink,J. (United States)
Sochor,J. (Czech Republic)
Sojka,E. (Czech Republic)
Solis,A. (Mexico)
Sondershaus,R. (Germany)
Sporka,A. (Czech Republic)
Stephane,R. (France)
Stich,T. (Germany)
Strengert,M. (Germany)
Stroud,I. (Switzerland)
Stylianou,G. (Cyprus)
Sumanta,P. (United States)
Szekely,G. (Switzerland)
Szirmay-Kalos,L. (Hungary)
Tang,W. (United Kingdom)
Taubin,G. (United States)
Teschner,M. (Germany)
Theußl,T. (Austria)
Torres,J. (Spain)
Ulbricht,C. (Austria)
Van Laerhoven,T. (Belgium)
Vanecek,P. (Czech Republic)
Velho,L. (Brazil)
Veltkamp,R. (Netherlands)
Vergeest,J. (Netherlands)
Viola,I. (Austria)
VOLLRATH,J. (Germany)
Vuorimaa,P. (Finland)
Wan,T. (United Kingdom)
Weidlich,A. (Austria)
Weiskopf,D. (Canada)
Westermann,R. (Germany)
Wood,J. (United Kingdom)
Wu,S. (Brazil)
Wuethrich,C. (Germany)
Yilmaz,T. (Turkey)
Zach,C. (Austria)
Zachmann,G. (Germany)
Zalik,B. (Slovenia)
Zambal,S. (Austria)
Zara,J. (Czech Republic)
Zemcik,P. (Czech Republic)

A Resolution-Independent Image Representation for Digital Cinema

Sylvain Brugnot Xiangyang Ju Paul Cockshott Paul Siebert
Computing Science Department, University of Glasgow, G12 8QQ, Glasgow, UK
{sylvain, xju, wpc, psiebert} @dcs.gla.ac.uk

ABSTRACT

We present a generic architecture for a novel type of image and movie representation designed to allow resolution-independent manipulation of movie sequences. By expressing the input signal as a sum of analytically-defined basis functions across space, time and scale, it becomes possible to compute image values at any location in the sequence, thereby providing a basis for both spatial and temporal resolution-independence. Distortion-constrained encoding of the basis function coefficients is furthermore built into the architecture to achieve data compression whilst minimizing signal degradation.

Keywords

resolution-independence, image coding, video encoding, post-production.

1. INTRODUCTION

Filmmakers today are increasingly turning towards an all-digital solution, from image capture to post-production and projection. Due to its fairly recent appearance, the digital cinema chain still suffers from limitations which can hamper the productivity and creativity of cinematographers and production companies.

Firstly, the variety of means by which digital material can be obtained (e.g., 35mm film scans, high-definition film cameras, video cameras) and the various display methods (TV, HDTV, projection) have led to the coexistence of multiple resolution standards, such as: PAL (720x576 pixels), 1080i (1920x1080 pixels), 2K (2048x1080 pixels) and more recently 4K (4096x2160 pixels). Similarly, this also occurs in the temporal domain, where values of 24, 30 or 60 frames/second coexist. The combination of these two factors results in over 25 different video formats. This absence of a unified standard of spatial and temporal resolution implies that digital film material from different sources is more costly to combine and composite at the production stage.

Secondly, the current standard for the storage and transfer of film data is the CINEON or DPX format (Digital Picture eXchange). Each DPX file encodes an individual frame as an array of uncompressed pixel values. Such files are extremely large, at typically over 10MB per frame, which corresponds to over a Terabyte per hour of film at 24 frames/second. Despite the ongoing technological advances in disk capacity and network speed, this makes the storage and transfer of movie data between collaborators along the digital cinema chain impractical and costly.

2. AN ARCHITECTURE FOR THE ENCODING OF MOVING IMAGES

In order to address the two issues mentioned above, we present an architecture for the resolution-independent encoding of digital cine material, which offers compression whilst minimizing resulting artifacts. This generic architecture allows a variety of specific implementations, each using different encoding schemes. The following concepts are the key to the architecture:

1. The cine sequence is treated as a three dimensional array of samples handled by voxel compression techniques.
2. For each block, a scale-space approach analogous to image pyramids is used to capture correlations across voxels in time and space.
3. At each level of the pyramid, the signal is expressed as a sum of analytically defined basis functions across space and time.

Posters proceedings ISBN 80-86943-04-6
WSCG'2006, January 30-February 3, 2006
Plzen, Czech Republic.

4. A closed loop feedback system is used to compensate for artifacts introduced by quantization at higher levels.

5. The quantization is constrained so that the resulting artifacts always remain within the statistical boundaries of in-camera shot noise.

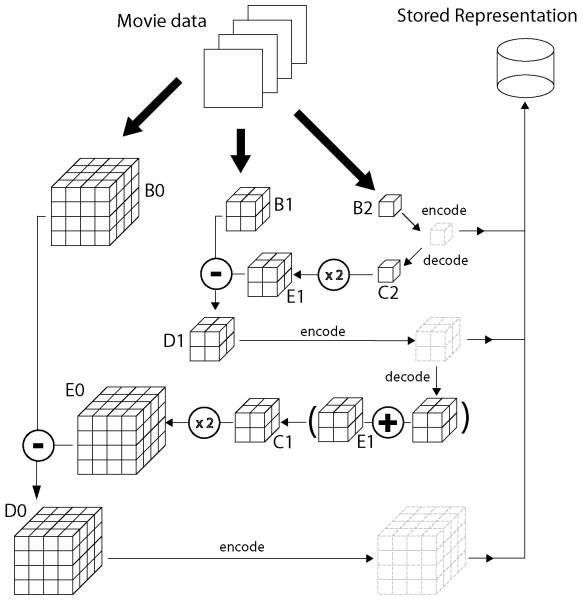


Figure 1. Encoding Architecture

The first phase of the algorithm, illustrated in Figure 1, gathers individual digital frames into three-dimensional blocks of pixels. Each block is then spatially and temporally filtered and downsampled repeatedly to form a structure where each successive block is smaller in each dimension by a scale factor *r* (typically 2 or 3). Blocks B0, B1 and B2 in Figure 1 form such a structure. We refer to this sequence of blocks as a pyramid by analogy with the commonly used image pyramids of [Bur83a]. We refer to the largest block as being at the base of the pyramid and the smallest as being at the top.

Consider the top block, B2 in the case of Figure 1. We apply to it some encoder (which, in the case of the top block, may be null). A copy of the encoded representation is sent to the output stream after which the encoded representation is internally decoded by the compressor and stored in block C2.

C2:= decode(encode(B2))

An interpolation algorithm, which may exploit features of the encoding previously applied, is then used to expand C2 by a scale factor of *r* to obtain block E1 which is the same size as block B1 but lacks high frequency information.

E1:=expandby(r,C2)

E1 may include some artifacts due to quantization or the interpolation technique used. We then form

D1:= B1-E1

so that D1 is a differential block containing the information from B1 that was not captured by the encoding of E1.

D1 is then itself encoded, perhaps using a different encoder than that used at the top level. The encoded output is appended to that produced for the top level of the pyramid. Again the encoded form of the current level is internally decoded and added to E1 to produce C1.

C1:= E1 + decode(encode(D1))

The block C1 should now be a close approximation to the original downsampled block B1. It can be expanded again by a factor of *r* to form a block E0, as large as the original source block B0. Subtracting them we again obtain a difference block:

E0:=expandby(r,C1)

D0:=B0-E0

which we encode, appending the ecoding to the output stream.

3. PRELIMINARY RESULTS

For evaluation purposes, the architecture presented here was implemented using trivariate polynomials as basis functions, resulting in an approach similar to the one found in [Cyg96a]. When no quantization scheme is applied, it was shown that the original signal can be reproduced with 100% accuracy, as well as rendered at varying resolutions with a result visually similar to that of well-established interpolation techniques such as bicubic resampling.

This architecture lends itself well to vector quantization methods, such as described in [Lin80a] and [Tao05a]. The performance of such a system is currently being investigated.

4. REFERENCES

[Bur83a] "The Laplacian Pyramid as a Compact Image Code", P.J. Burt, E.H. Adelson, IEEE Transactions on Communications, vol. 31, n. 4, 1983, pp 532-540

[Cyg96a] "3-D Block Decomposition Approach for Video Coding", Boguslaw Cyganek, Image Processing & Communications, An International Journal, 1996, vol. 2, n. 4

[Lin80a] "An Algorithm for Vector Quantizer Design" Y. Linde, A. Buzo, et al., IEE Transactions on Communication, 1980, vol. 28, pp. 84-95

[Tao05a] "Distortion-Constrained Compression of Three-Dimensional CLSM Images Using Image Pyramid and Vector Quantization", Y. Tao, PhD Thesis, University of Glasgow, 2005.

Matching for Perceptual User Interface

Jose Maria Buades Rubio
Universitat Illes Balears
Ed. Anselm Turmeda
CrtaValldemossa Km 7.4
Spain (E-07122), Palma Mca
josemaria.buades@uib.es

Francisco Perales López
Universitat Illes Balears
Ed. Anselm Turmeda
CrtaValldemossa Km 7.4
Spain (E-07122), Palma Mca
paco.perales@uib.es

Javier Varona Gómez
Universitat Illes Balears
Ed. Anselm Turmeda
CrtaValldemossa Km 7.4
Spain (E-07122), Palma Mca
xavi.varona@uib.es

ABSTRACT

In this paper we describe a complete method for building a perceptual user interface in indoor uncontrolled environments. Overall system uses two calibrated cameras and does initialization: detects user, takes his/her measurements, builds a 3D-Model; and also performs matching/tracking for: trunk, head, left arm, right arm and hands.

Keywords

Human Computer Interaction, Perceptual User Interface, Matching, Virtual Reality Interaction

1. INTRODUCTION

Overall system uses two calibrated cameras and does initialization: detects user, takes his/her measurements, builds a 3D-Model; and also performs matching/tracking for: trunk, head, left arm, right arm and hands. System is waiting for a user in a predefined posture, once user has been detected he/she is analysed to take measurements and build a 3D-Model. Tracking is carried out by a Montecarlo probabilistic method and divided in three steps, track trunk and head, left arm and right arm, this divide and conquer solution proposed improve computation time without getting worse results. Matching process uses two sub-matching functions, one to compute colour seemed and another to compute shape one.

2. INITIALIZATION

Initialization is performed from one camera. After user appears in action system detects him [Bua03] and takes measurements for later matching process, as shown in figure 1, system models trunk region as a rectangle, hands, arms and head regions as 3D-ellipsoids. 3D reconstruction is performed taking into account that user is at same distance that calibration object was at calibration process.

Posters proceedings ISBN 80-86943-04-6
WSCG'2006, January 30-February 3, 2006
Plzen, Czech Republic.

Figure 1. User modeling from single image.

3. MATCHING

The matching process starts in the first frame after user detection, and it is performed taking account the following visual cues: color and shape. Besides, we will use spatial constraints.

Tracking is performed using a Montecarlo strategy in two steps. First, the system try to match the trunk and head segments. After, the arms (left and right separately) are matched using the previously computed trunk position. These two steps reduce the computing time, instead of perform a global search, we benefits from divide and conquer strategy.

Search Strategy

As we have commented before, we exploit the benefits of a divide and conquer strategy. The trunk segment area is the large region and never is occluded. For this reason, this strategy obtains good results; else the results would be unexpected positions when an occlusion occurs. Second step is to perform matching for the most interesting body parts, the hands. For each arm is performed a Montecarlo search in an independent way. This three matching process (trunk and head – left arm – right arm) reduces computation, instead of multiply computation time for trunk, left and right arm, computation times are added. The results are quite

good as it is in the results section. Proposed positions are evaluated as biometrically possible to avoid undesirable results and reduce computation time.

Matching Function

The matching function is composed of two parts: colour comparison and shape comparison.

The trunk segment is modelled as a 2D rectangle, others shapes as a 3D box have been tested but surprising the results are worse. The rest of segments (arm, forearm, hand and head) are modelled using super-ellipsoids.

For each segment, the colour model is computed from initialization process (a better solution would be to have a texture model). The segment is projected in the image to evaluate a colour matching function; each pixel is compared with the colour model and scored as good (1) or bad pixel (0), for trunk segment only are projected a grid of 15x10 pixels with so good results as projecting all the segment. The other segments, super-ellipsoids, also a sample of pixels are projected. For each segment colour matching function returns the ratio of good pixels detected.

The shape matching function uses the contours. To detect contour in the captured image performs Sobel operand, as a result we get an image, called Sobel-image. Segment contours are projected in Sobel-image and scored pixels as good (1) or bad (0) in relation with its value. This function returns the percentage of pixels scored as good. This function has good results if the segment is just in the contour of the image, and it is very difficult because the real person shape is very deformable. A better solution is to take into account the distance from the segment contour to the contour in the Sobel-image. In this second solution not only the exact pixel is compared, neighbour pixel in normal direction (from near to far) are evaluated, thus a pixel is evaluated as a value between 0 and 1, in relation to the distance to a contour in sobel-image, more near a greater value.

Matching function returns a value for a segment s_j as follows:

$$m(s_j) = \min_{\forall \gamma_i \in H} (m^{\gamma_i}_{colour}(s_j) + m^{\gamma_i}_{shape}(s_j)) \quad (1)$$

where H is a set of cameras, $\{\gamma_1,\gamma_2\}$ in this case, s is a segment, $m^{\gamma_i}_{colour}(s)$ is colour matching function from camera γ_i, $m^{\gamma_i}_{shape}(s)$ is shape matching function.

Finally, the Matching value for a pose Ψ is defined in Equation (2).

$$m(\Psi) = \sum_{j=1}^{N} m(s_j) \quad (2)$$

4. ACKNOWLEDGMENTS

This work has been subsidized by Universitat Illes Balears, and the national project TIN2004-07926 from the MCYT Spanish Government. Acknowledges the support of a Ramon y Cajal fellowship from the Spanish MEC.

5. REFERENCES

[Bal96] C. Ballester, V. Caselles and M. Gonzalez, "Affine invariant segmentation by variational methods", SIAM J. Appl. Math., Vol. 56, No 1, pp. 294-325, 1996

[Har00] I. Haritaoglu, "W4: Real-Time Surveillance of People and Their Activities" IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 22 No8, pp 809-830, 2000

[Sid00] H. Sidenbladh, M.J. Black and D.J. Fleet "Stochastic Tracking of 3D Human Figures Using 2D Image Motion" ECCV 2000.

[Bua03] J.M. Buades, M. Gonzalez, F.J. Perales. "A New Method for Detection and Initial Pose Estimation based on Mumford-Shah Segmentation Functional". IbPRIA 2003. Port d'Andratx. Spain. June 2003. pp 117-125

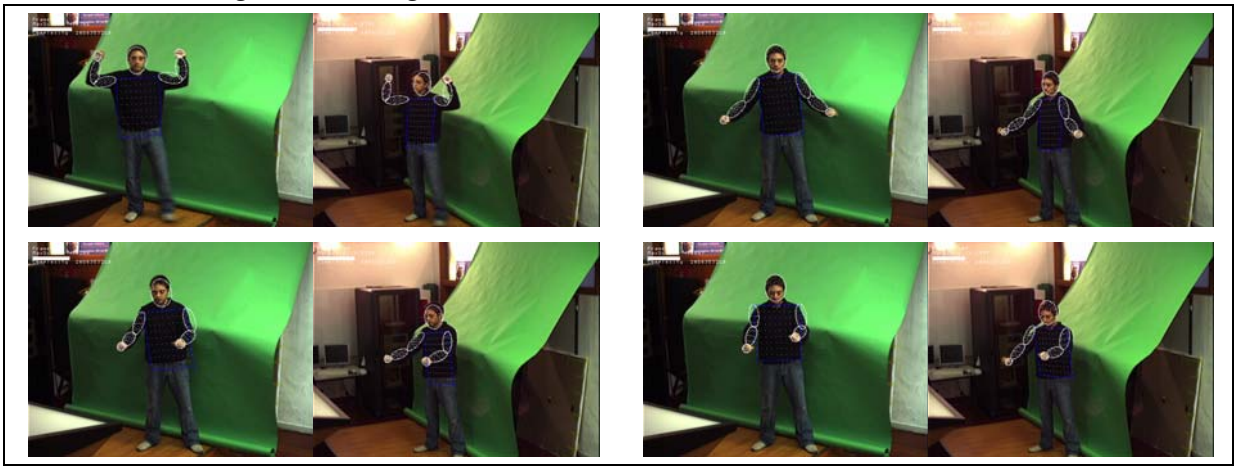


Figure 2. Tracking results.

Fast algorithm for cloud rendering using flat „3D textures”

Dariusz Domański
Institute of Computer Science,
Warsaw University of Technology,
ul. Nowowiejska 15/19,
00-665 Warsaw, Poland
ddomansk@ii.pw.edu.pl

Abstract

This paper presents an efficient method to render clouds in their natural dynamic (*i.e.*forming, dissipating, flowing). The algorithm uses flat „3D textures” to represent volumetric data. This is used to compute the amount of light reaching each voxel. The light values are computed in real time and then visualised using a slicing technique to enable on-line interaction with the cloud model.

Keywords: natural phenomena visualization, 3D rendering, real-time animation

1 INTRODUCTION

Clouds are often included in online applications such as flight simulators and video games. The problem is that rendering clouds in a fast and realistic way is very complex. Various simplifications are used to allow the real-time rendering of cloudy scenes, however very often they decrease the quality of the animation. A very valuable objective is to be able to produce scenarios where the observer is flying through clouds.

In this paper, simple cellular automata was used to obtain cloud data in a few processing steps. These data are then processed with the proposed cloud volume lighting algorithm and visualised with a technique using slices to approximate to the volume.

2 PREVIOUS WORK

The most realistic cloud rendering method, so far produced, was developed by Harris and Lastra [HL01]. The algorithm takes into account the phenomena of light scattering: multiple forward scattering along the light direction and single scattering towards the viewer.

This work was extended by Harris *et al.*[HBSL03] to model realistic motion of clouds on GPU via stable fluid simulation introduced by Jos Stam. The improved algorithm deals with the rendering aspect of the problem as they mentioned a method for rendering in 3D condensed vapour density provided by simulation. This extension is discussed in more detail by Harris in his PhD dissertation [Har03].

Posters proceedings, ISBN 80-86943-04-6
WSCG'2006, January 30 – February 3, 2006
Plzen, Czech Republic.

Light scattering was introduced even earlier by Dobashi *et al.*[DKY+00], however, theirs method modelled single scattering only. A common feature of all these rendering algorithms is that they occur in two phases. The first pass calculates the scattering along the light vector and creates the lighting information. The lighting data is then applied to the cloud model in the second phase of the algorithm.

A simpler approach was used by Vane [Van04] based on work by Behrens and Ratering (widely cited in Vane's paper). The visual effect of his work demonstrates that Behren's algorithm is suitable for shadowing but not for lighting.

3 VOLUMETRIC DATA

For the proposed rendering method clouds are represented as density distribution model. From the array of cloud data generation methods the Dobayashi's algorithm was selected (described in detail in [DKY+00]).

The computationally expensive process of the continuous distribution calculation (called *smoothing*) was spread over a number of animation frames. The assumption given for real-time applications is that each frame must be displayed in 40 milliseconds there is about half this time available for simulation or smoothing with visualization technique proposed in section 4.

4 VISUALIZATION MODEL

The most common approach to visualize volumetric data is to use a slicing algorithm. The basic idea is to take a slice of volume data, render it onto texture and paste this texture on quads in 3D space. Figure 1 shows mapping of volumetric data onto a flat „3D texture” which is used to cover the slice's surface.

As denoted by Vane in [Van04] there are two classes of slicing algorithms: object aligned and view aligned. For the implementation of the visualisation model object aligned slices was chosen. The key advantages of

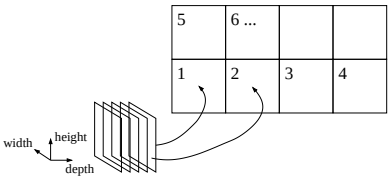


Figure 1: Mapping volumetric data of discrete space onto flat „3D texture”

this method are its efficiency and its simplicity, considered both methods introduce similar artifacts: when the viewer is very close (or inside) the volume the space between the planes becomes apparent and the whole object can be perceived as a stack of slices.

5 RENDERING

The rendering algorithm proposed in this section is similar to the original concept of Harris and Lastra [HL01]. The key difference is that the method described below adopts images of volume rendered in slices and then uses hardware blending function to process the whole slice simultaneously. This approach speeds up the whole process as the additional buffer used to read back the illumination for each pixel can be read as one operation for whole slice.

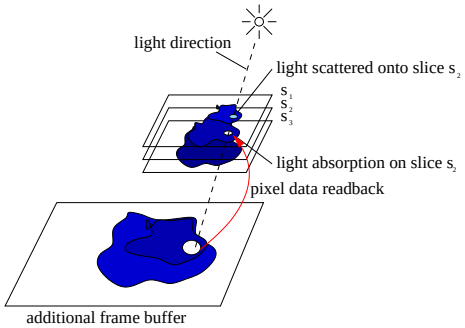


Figure 2: General idea for cloud illumination algorithm using slices: illuminating slice s_2

Flat „3D textures” illuminated by presented algorithm 1 (picture 2) can now be displayed with visualization technique described in section 4.

6 CONCLUSIONS AND FUTURE WORK

The sample results included in this paper (picture 3) were actually generated by a real time application running at 30*fps* and better.

The proposed lighting algorithm works well for reasonable sizes of volume data and numbers of slices (experiments were done with volume $64 \times 64 \times 64$ with 64 slices). Further extensions, which will enable the natural simulation of day lit skies and rendering of clouds in a large scale space will add significant realism to animated scenes.

Algorithm 1: Cloud lighting algorithm

input: model viewed align its axes $\vec{X}$, $\vec{Y}$ and $\vec{Z}$ (three sets of slices covered with textures), normalized vector of light direction $\vec{l}$, global light color c, scattering coeficient a, absorption factor e

/* determine the set of slices to be used for lighting and the order they are processed */

1 `GetMaxABSVecIndex(`$\vec{l}\cdot\vec{X}, \vec{l}\cdot\vec{Y}, \vec{l}\cdot\vec{Z}$`)`
2 `CreateAndFillBuffer(`$b, res_w \times res_h, c$`)`
3 **foreach** slice s **do**
4 **foreach** pixel $p_{(u,v)}$ **do**
  /* compute pixel coordinates and radius corresponding to buffer content */
5 $(u_b, v_b, r) \leftarrow$ `ProjectPixel(`$b, p_{(u,v)}$`)`
6 $\tau \leftarrow \sum_{u'_b=-\frac{r}{2}}^{\frac{r}{2}} \sum_{v'_b=-\frac{r}{2}}^{\frac{r}{2}}$
7 `GetPixelAlpha(`$b, p_{(u_b+u'_b, v_b+v'_b)}$`)`
8 $\tau_{(u,v)} \leftarrow \tau \cdot \frac{a \cdot e}{4 \cdot \pi^2 \cdot r^2}$
9 `SetPixelColor(`$s, p_{(u,v)}, \tau_{(u,v)} \cdot c_{(u,v)}$`)`
10 `DrawSlice(`b, s`)`

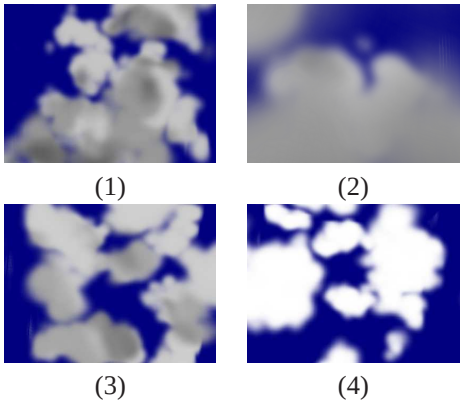


Figure 3: Sample frames from cloud simulation: (1) and (2) show same cloud model viewed from different perspectives; (3) experiments with increasing scattering factor; (4) unlit cloud model (very bright day)

REFERENCES

[DKY+00] Y. Dobashi, K. Kaneda, H. Yamashita, T .Okita, and T. Nishita. A simple, efficient method for realistic animation of clouds. In *ACM SIGGRAPH*, pages 19–28, 2000.

[Har03] M. J. Harris. *Real-Time Cloud Simulation and Rendering*. PhD thesis, Department of Computer Science, University of North Carolina at Chapel Hill, 2003.

[HBSL03] M. J. Harris, W. V. Baxter, T. Scheuermann, and A. Lastra. Simulation of cloud dynamics on graphics hardware. In *Graphics Hardware*, 2003.

[HL01] M. J. Harris and A. Lastra. Real-time cloud rendering. *Computer Graphics Forum*, 20(3):76–84, 2001.

[Van04] E. Vane. Cloud rendering using 3d textures, 2004.

Measurement of Medical Parameters Based on 3D Surface Models

Pawel Drapikowski
Poznan University of Technology
Institute of Control and Information Engineering
ul. Piotrowo 3a, 60-965 Poznan, Poland
pdr@ar-kari.put.poznan.pl

ABSTRACT

In medical diagnostics one need to perform quantitative analysis and measurements of 2D or 3D data. The length, angle, area of region, 3D surface area, volume are measure to determine medical parameters. This paper presents uncertainty estimation of 3D surface model created from object boundaries using CT, MRI series of images. Error propagation from CT images acquisition to the anatomical structure pointed (measured) by operator using 3D model cross-sections is introduced.

Keywords

3D surface model, uncertainty estimation.

1 INTRODUCTION

In medical diagnostics, mainly in surgery, orthopaedic surgery, neurosurgery, it is necessary to estimate many geometrical parameters: length in 2D and 3D space, angle, object volume and surface area. Values of these parameters are helpful while distinguishing between the pathological and normal condition of the human body, and estimating the degree of pathology. Monitoring values of different parameters during the treatment allows for a quantitative analysis of the improvement or deterioration of diseases. The measurements can be done on the base of 2D radiological images, CT or MRI slices and 3D surface or volume models. 3D surface and volume models are also used for pre-operative planning and intra-operative navigation. There is a lot of computer software that can create 3D surface and volume models based on 2D cross-section images and measure different parameters. However, none of them enables the metrological analysis of the obtained results of measurements. The validation study of the accuracy of medical modelling based on a semi-anthropomorphic phantom of human head was carried out within the scope of PHIDIAS project. The measurements of the phantom and its model (built using RPT (Rapid Prototyping Technology) of a geometrical surface model created of 2D cross-section CT images of the phantom) were carried out using a coordinate measuring machine. The measured characteristics of the phantom and the model were compared.

Posters proceedings ISBN 80-86943-04-6
WSCG'2005, January 31-February 4, 2006
Plzen, Czech Republic.

2 MOTIVATION

The analysis of measurement error in diagnostic imaging is a key problem to ensure the reliable results of measurements. It contributes to convince physicians to apply modern hardware and software tools in diagnostics. The quantitative error estimation of 3D model is also very important in pre-operative planning and intra-operative navigation, where the accuracy of the model and measurements is critical.

3 ACCURACY ANALYSIS FOR 3D SURFACE MODELS CREATION

The accuracy of the model is dependent on a lot of factors: resolution and processing of 3D image data, thickness and distance between slices, the method of reconstruction (control parameters of decimation algorithm and smoothing filter), factors connected with marking measurement points by the operator. To determine the accuracy of 3D surface model, the following procedure was developed. The CAD model had a geometrical structure similar to the human costal geometrical structure (curves with various radii, openings of various diameters, sharp edges) and the internal density similar to the internal density of bones. The CAD model was scanned with a CT scanner with a spiral scan and slice thickness equal to 1mm. The image data was processed (threshholding, filtering). A 3D surface model was created using the marching cube algorithm.

3.1 Analysis of image data

The first step of the error analysis was the comparison of image data with cross-sections of the CAD model (contour images). Image data were contoured and the longest contour was determined (the following contours were skipped). The same longest contour was determined on cross-sections of the CAD model. Both contours were scaled to the same scale and placed in two raster images, filling the interior of each contour. The XOR Boolean operation was performed to yield the third image, which contained the shape disparity of both contours. The comparison method of two images is called Disparity Surface Method (DSM).

The error of single contour comparison e_{pk} is:

$$e_{pk} = \frac{P_r * O_p}{O_k * P_p} 100\% \qquad (1)$$

where:

P_r - surface area of disparity image

P_p - surface area of the rectangle circumscribed on the contour

O_p - periphery of the rectangle

O_k - length of contour

The fraction O_p/O_k in formula 1 allows to correct the estimated percentage error in the situation where the surface area of the filled contour is small in relation to the length of the contour. The comparison was carried out with the distance between images 1mm, equal to CT slice thickness.

3.2 Analysis of the 3D surface model

The error estimation of the 3D surface model was performed by means of comparing the CAD model with the reconstructed model. The longest contours of the cross-sections of both models were compared using the DSM method with the distance between 0.5mm slice (CT slice thickness was 1.0mm). The orientation of the cutting plane was perpendicular and parallel to the scanning direction. The error of the whole model e_m is the mean error of all cross-sections. The maximum percentage error of the model is the maximum percentage error of the cross-section.

3.3 Measurement verification

The measurement verification of medical parameters using the reconstructed 3D model is impossible because there is no access to actual anatomical structures in the patient. Therefore, the experiment was performed that consisted in the measurement of the same geometric values on the CAD model and on the reconstructed CAD model. The measuring procedure was similar to the measuring procedure of measurement performed for the dysplastic pelvis in order to establish orthopaedic parameters, and it was performed in the following way: within the openings of the tested model, spheres were placed in such a way that the sphere surface closely adheres to the inner surface of the opening, a line was drawn through the centres of the spheres, the angles were established, whose apices were situated in the centres of the spheres, one of arms was the line, while the other arm crosses the specific points of the model.

4 RESULTS AND CONCLUSIONS

The comparison results of CT slices and CAD model, within the range of the mean percentage error and the maximum percentage error, are as follows:

$$e_m = 0.22\% \qquad e_{max} = 0.28\%$$

The mean percentage errors and maximum errors in the direction of each axis for reconstructed 3D surface model present as follows:

	Mean error [%]	Max. error [%]
Z axis	2.54	7.4
Y axis	4.08	15.2
X axis	2.90	20.7
Entire model	3.17	20.7

The slightest reconstruction errors occur always in the direction of CT slice image registering. The error analyse suggest that the model error is not constant and it depends on the surface shape and scanning direction. It may easily be seen that the most serious errors of the reconstructed model occur always in places where a sudden change of surface shape appears, and the normal for the surface is directed at a large angle in relation to the scanning direction.

The results of measuring verification according to the procedure mentioned above are in the range of 0.4-0.8%. Moreover, the standard deviation is rather small, which suggests considerable repeatability of measurement.

Analysing the procedure of processing CT image data and CT images for the 3D surface model and the measurements performed on this model, it may be stated that the largest error occurs in the process of the 3D surface model formation (see below). Strangely enough, the error of measurements performed on this model is considerably smaller than the error of the model itself. The explanation is as follows: measurement points, used for the calculation of angle values, and the distance, were reconstructed accurately, while the error of the model is the most strongly influenced by the inaccuracy of surface reconstruction in places where a sudden distance change occurs and the slope angle of the normal towards the surface in relation to the scanning direction is large.

CT Slice error - 0.22%
3D surface model error - 3.17%
Measurements of 3D model error - 0.69%

A Simple Anti-Aliasing Method For Straight Line Drawing Based On DSP Platform

Dusheng Wang
Institute of Electromechanical & Control Engineering
Dalian Nationalities University
Dalian Development Zone, Liao He West Road 18
Dalian City, Liaoning Province, P.R.China, 116600
dusheng3791@hotmail.com

Hock Lye Toh, Xiang Chen, Fan Yang

Centre for Signal Processing
Nanyang Technological University
Nanyang Drive 637722, Singapore
{ ehltoh, exchen, efyang}@ntu.edu.sg

ABSTRACT

A simple and practical anti-aliasing method for a color straight line drawing is presented in this paper. The method has been applied in a DSP-based display system to remove the undesired jaggies occurred in the line drawing. The experimental results show that this method can produce a good visual effect on the low resolution display screen.

Keywords: anti-aliasing, straight line drawing, DSP-based system, raster graphics

1. INTRODUCTION

When we try to draw a line on a low resolution screen , we are certainly to see the ugly jagged "stair-steps", this is because the screen does not have a enough resolution to display a line smoothly. Anti-aliasing is a way to use color information to make up for a lack of screen resolution. Among the existed methods, Wu's algorithm[Wu91] is mostly used and efficient. When the background is black, the anti-aliasing effect with Wu's method is rather good in visual aspect. But when the background is not black, the visual effect will become worse. This paper presents a modified anti-aliasing algorithm based on Bresenham's algorithm. It deployed the Wu's idea that draws the two pixels bracketing the line along the minor axis at each point. But each pixel's drawing intensity is assigned the weighting combination of drawing color and background color at that point. Section 2 will describe our proposed method in details. Section 3 and 4 are experimental result and conclusion.

2. PROPOSED METHOD

Our proposed method for anti-aliasing is based on Bresenham's line drawing algorithm. When an x-majored line is plotted using it, all pixels that are overlapped by the line area are displayed with an intensity proportional to the area of overlap Instead of plotting the line with a single pixel at each x position. There are various ways to calculate the overlapping area[Gupta81, Foley90]. In our approach, a simplified technique is adopted. For a x-majored line, on each column, the line overlaps at most two pixels. The ratio of these two overlapped areas is proportional to the ratio of (1- D_1) and (1- D_2), as depicted in Fig. 1.

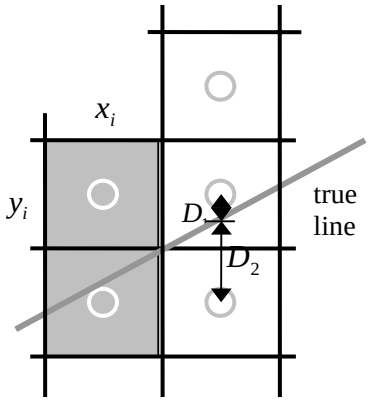


Fig. 1 Vertical distance of measurement for a x-majored line

To satisfy the requirement of change in background color, we choose the weighting combination between drawing color and background color, that is, pixel ($x_i + 1, y_i$) will be assigned an intensity of being equal to the sum of (1- D_1) times intensity of drawing color and D_1 times that of background color; while

Posters proceedings ISBN 80-86943-04-6
WSCG'2006, January 30-February 3, 2006
Plzen, Czech Republic.

$(x_i + 1, y_i - 1)$ will be assigned an intensity of being equal to the sum of (1- D_2) times intensity of drawing color and D_2 times that of background color. For y-majored line, this intensity division will be applied to two neighboring pixels bracketing it in a row. For the RGB color line, the following equations express clearly the weighting intensity operation on it:

$$\begin{aligned} R &= (1 - D_1) * R_d + D_1 * R_b \\ G &= (1 - D_1) * G_d + D_1 * G_b \qquad for\ (x_i + 1, y_i) \\ B &= (1 - D_1) * B_d + D_1 * B_b \end{aligned}$$

$$\begin{aligned} R &= (1 - D_2) * R_d + D_2 * R_b \\ G &= (1 - D_2) * G_d + D_2 * G_b \qquad for\ (x_i + 1, y_i - 1) \\ B &= (1 - D_2) * B_d + D_2 * B_b \end{aligned}$$

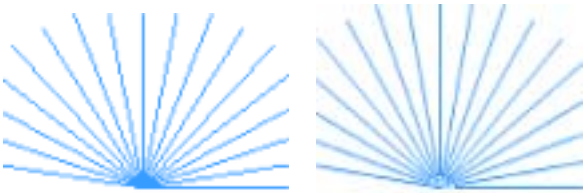
where R, G, B is the resulting red, green, blue component of the drawing point, R_d, G_d, B_d is the red, green, blue component of the drawing color, while R_b, G_b, B_b is the red, green, blue component of the background, and also it exists $D_1 + D_2 = 1$ in the distance measurement.

3. EXPERIMENTAL RESULTS

Based on DSP(SHARC ADSP21060 chip) platform, we developed a Video Graphics System with the support of dedicated OpenGL[Neider93] driver. The most common 2-Dimension related basic OpenGL functions had been implemented on this platform. Drawing line function was finished based on Bresenham's algorithm, and our proposed anti-aliasing method was added to remove the jagged "stair-step" in the straight line drawn. In our raster display screen with 640×480 resolution, some graphic objects would be required to be displayed on the video picture or different intensity level of background. When we draw a line on the white background and use Wu's anti-aliasing method, the anti-aliasing effect looks like as Fig.2 shown, Fig.2 is depicted as follows: a line with the length of 60 pixels was rotating around the end point at the center of screen at the interval of 10 degree every frame. We got 18-19 frames' data from the display buffer in the real-time running simulation platform and plotted it with MATLAB.

As the background's white color is not used to do color mixing, the line looks like a dirty "stair-step" edge. Fig.3 depicts the effect with our proposed anti-aliasing method for a straight RGB color line. The data collection and plotting procedures are the same as that in Fig.2. Fig. 3(a) shows different dimensions lines without anti-aliasing. Fig. 3(b) demonstrates different dimensions lines with anti-aliasing.

Fig. 2 The example using Wu's anti-aliasing method in white background



(a) The figure without anti-aliasing (b)The figure with anti-aliasing

Fig. 3 The single-pixel-line with different dimensions drawn by Bresenham's algorithm

4. CONCLUSION

This paper presents a simple and practical anti-aliasing method for a color straight line drawing. From the experimental result we can draw the conclusion that our anti-aliasing method is more adaptable for anti-aliasing RGB color line than Wu's method and produces a good visual effect on the low resolution screen. But it is a time-consuming function too. It takes about 60 cycles on DSP platform to write an anti-aliased pixel's color information into display buffer while writing a pixel color information without anti-aliasing is only needed to take one cycle. Therefore, implementing anti-aliasing function had better been finished by hardware module if necessary.

5. REFERENCES

[Foley90] J. D. Foley, A. van Dam, S. K. Feiner, J. F. Hughes: *Computer Graphics: Principles and Practice, Second Edition*. Addison-Wesley, Reading 1990.

[Gupta81] S. Gupta, R. E. Sproull: Filtering Edges for Gray-Scale Displays. *Computer Graphics* 15 (3) (*Proceedings of SIGGRAPH'81*),August 1981, ACM SIGGRAPH New York, pp. 1-5.

[Neider93] J. Neider, T. Davis, and M. Woo: *OpenGL Programming Guide*. Addison-Wesley, Menlo Park, 1993.

[Wu91] X. Wu: An Efficient Antialiasing Technique. *Computer Graphics*, 25 (4) (*Proceedings of SIGGRAPH'91*), July 1991, ACM SIGGRAPH New York, pp. 143-152.

Parametric Discretization of Supercover Simplex

Allan Fousse, Eric Andres
SIC Laboratory, bât. SP2MI
Bd Marie et Pierre Curie, BP 30179
86962 Futuroscope Chasseneuil Cedex, France
{fousse,andres}@sic.univ-poitiers.fr

ABSTRACT

Within the framework of the development of a software which aims at unifying geometrical modelisation and image processing, based on both analytical and voxels representation of objects, we wish to extend this to parametrical objects. We focus, in this paper, on the study of parametrical simplexes and their relation with the discrete supercover model. Besides, we obtain a simple and incremental algorithm to run over n-D discrete objects. This is also the basis of parametrical curves and surfaces studies.

Keywords

Discrete geometry, geometrical and topological modelling, parametric rasterization.

1. INTRODUCTION

2D and 3D geometrical modelling and image processing do not use similar tools. Geometrical modelling needs high level informations such as topological and geometrical models, although image processing uses low level informations like pixels and voxels. In many applications, we would like to use both type of representations at the same time, for example virtual scalpels in a discrete image resulting of a scanner to simulate a medical operation. For a few years, a discrete modeler called SPAMOD (for *Spatial Modeler*) is being developed [DDLA05]. The purpose of this modeler is to permit, in the same software, to have a representation of an object in an Euclidean form together with a discrete form. The various forms (embeddings) of the object are hold in a coherent way using a hierarchical structure based on topology.

The hierarchical structure of SPAMOD, on one hand, a geometrical modeler based on topology. On the other hand, on a discrete analytical model, the supercover model [ANF97], sets up the pixels and voxels. In a discrete analytical model, we can describe a discrete object with a number of inequations which do not depend on the total number of pixels that compose the object. So, for the supercover model, a 3D triangle of whatever size is described by 17 inequations at the most. In fact, these inequations represent the convex hull of the voxels, and this convex hull is connected to the boundary representation of the Euclidean geometrical models.

The purpose of our study is to provide an appropriate parametrical description of supercover objects unlike [Kau87]. The first step, which is developed in this paper, aims at proposing a parametrical description of the supercover simplexes thanks to a parametrical method and also its discretisation algorithms.

First, in Section 2, with the example of a 2D segment, we describe the principle on which the parametrical description of a simplex works. We extend our analysis to simplexes of higher dimensions in Section 3, and conclude in Section 4.

Posters proceedings ISBN 80-86943-04-6
WSCG'2006, January 30-February 3, 2006
Plzen, Czech Republic.

2. A PARAMETRICAL APPROACH OF THE SEGMENTS

We focus, in a first step, on the 1-dimensional simplex (straight line segment). Let be a segment defined by two points P and Q which have these coordinates (P_x, P_y) and (Q_x, Q_y) respectively. We will study this problem with a parametrical approach. The segment PQ can also be represented by a piece of parametrical line, such as $\vec{PX} = t\ \vec{PQ}$ with t between 0 and 1 :

$$D(x,y) = \begin{cases} bt \\ at \end{cases}, \text{ with } t \in [0,1]$$

We focus on the supercover discretization of points

which have coordinates (bt, at) with t between 0 and 1. This parametrical formula has the advantage to show that the incremental step on X (respectively on Y) is only linked with the parameter t and the constant value b (respectively a).
Our goal is to characterise the values of t for which an incremental step on X or on Y is defined. To do this, we can compute the value of t where the segment cuts a pixel border, that is a point with one half-integer coordinate. We deduce a series S_x (respectively S_y) of parameters t where we have an incrementation on X (respectively Y) : $S_x(k)$ series of t such as

$$bt = k + \frac{1}{2} \Rightarrow t = \frac{2k+1}{2b} \text{ for } k \geq 0$$

and $S_y(k)$ series of t such as

$$at = k + \frac{1}{2} \Rightarrow t = \frac{2k+1}{2a} \text{ for } k \geq 0$$

Now, we only have to sort the two series S_x and S_y to know what order of incrementations on X or on Y we have to make so as to draw the line.
For limiting the line to the segment PQ, we only need to restrict t between 0 and 1, that is we only need to take values of the two series S_x and S_y with k integer from 0 to $b-1$ (respectively $a-1$) for S_x (respectively S_y).
This construction is directly extendable in 3D. We only have to sort not two series but three S_x, S_y and S_z.

3. SIMPLEXES OF HIGHER DIMENSIONS

Now, we extend this principle to the 2D simplex: the triangle. Let P_1, P_2 and P_3 be three points with coordinates (X_1, Y_1), (X_2, Y_2) et (X_3, Y_3) respectively. A point U belongs to the triangle if and only if we can write: $\vec{P_1U} = t\,\vec{P_1P_2} + t'\,\vec{P_1P_3}$ with $t, t', t+t' \in [0,1]$.
We use a parametrical representation to obtain a new supercover discretisation algorithm. With A_1 and B_1 (resp. A_2 and B_2) are the directrix coefficients of the line which contains the segment P_1P_2 (resp. P_1P_3) we have :

$$T(x,y) = \begin{cases} A_1t + A_2t' \\ B_1t + B_2t' \end{cases} \text{, with } t, t' \in [0,1]$$

From these equations, we can deduce an incremental step on X (respectively on Y) when $A_1t + A_2t' = \frac{2k+1}{2}$ (respectively $B_1t + B_2t' = \frac{2k+1}{2}$) for $k \geq 0$. We deduce the two following series S_x and S_y :

$$S_x(k) = \{t, t'\} \text{ such as } A_1t + A_2t' = \frac{2k+1}{2} \text{ for } k \geq 0$$

and

$$S_y(k) = \{t, t'\} \text{ such as } B_1t + B_2t' = \frac{2k+1}{2} \text{ for } k \geq 0$$

With a geometrical interpretation of these two series, in a space built by the parameters t and t', we obtain that S_x (respectively S_y) is nothing else but the series of parallel lines such as $A_1t + A_2t' = \frac{2k+1}{2}$ (respectively $B_1t + B_2t' = \frac{2k+1}{2}$) for $k \geq 0$.
If we study the relative position of these two sets of lines. We can find what incremental step we must do without explicitly computing the values of t et t', using only a comparison.
As for segments, we do not need to work with two series but three to obtain a 3D incremental algorithm of supercover discretisation for a discrete triangle.
In practice, the algorithm is based on a notion of order built on parameters t and t', providing a traversal order on the pixels or voxels of the triangle. This order provides a direct generation of all voxels instead of an algorithm based on 2D-filling and then retro-projection in 3D space.
It is also possible to apply the same process for a 3D simplex (tetrahedron). With the same principle, we have to study the relative position of three pieces of planes, which can be computed by many differents methods [EOS86]. For higher dimensions, we only need to check if the geometrical objects we need to handle, always allow us to extend our algorithm.

4. CONCLUSION

In this paper, we show, for the supercover model, a parametrical representation of the supercover of various discrete simplexes. It provides a natural order for the pixels, voxels of simplexes. In discrete modelling, we often have to run over a discrete object so as to realise various operation. As soon as we do not handle a line segment, we loose the notion of order. In the higher dimensions, coverage algorithms are often complicated. In this particular application we hope that parametrical description could help.

5. REFERENCES

[ANF97] E. Andres, P. Nehlig, and J. Francon. Tunnel-free supercover 3d polygons and polyhedra. *Computer Graphics Forum*, 16(3):C1–C13, 1997.

[DDLA05] G. Damiand, M. Dexet, P. Lienhardt, and E. Andres. Removal and contraction operations to define combinatorial pyramids : application to the design of a spatial modeler. *Image and Vision Computing*, 23:259–269, 2005.

[EOS86] H. Edelsbrunner, J. O'Rourke, and R. Seidel. Constructing arrangements of lines and hyperplanes with applications. *Society for Industrial and Applied Mathematics Journal on Computing*, 15:341–363, 1986.

[Kau87] A. Kaufman. Efficient algorithms for 3d scan-conversion of parametric curves, surfaces and volumes. *SIGGRAPH 87, Anaheim, USA, Comp. Graphics Forum*, 21(4):171–179, 1987.

$\sqrt{3}$ Subdivision and 3 connected meshes with creases, boundaries and holes.

Olivier Guillot **Jean-Paul Gourret**

Laboratoire Informatique, Image, Interaction (L3i) Université de La Rochelle
IUT department d'informatique
15 rue François de Vaux de Foletier
17026 La Rochelle Cedex 1
olivier.guillot@univ-lr.fr jean-paul.gourret@univ-lr.fr

ABSTRACT

A project of surface meshing for still and animated images and associated software has been developed at L3i for several years. The software ([KG04]) implements a meshing technique of deformable objects based on triangular meshes or on polygons with only 3 connected vertices. In this paper, we present a technique of subdivision of triangular meshes and its adaptation to 3 connected meshes. The technique insures Cn continuity (n = 1 or 2), takes into account discontinuities such as creases, darts and boundaries, and allows approximation or interpolation of surfaces.

Keywords

Mesh – subdivision - surface.

1. INTRODUCTION

The subdivision is based on the insertion of a new vertex in each triangular face. For several years we have been interested by deformable meshes based on physical laws. Then we have introduced 3 connected meshes, because deformation calculations are faster than with other kind of meshes. The rapidity is due to the minimal number of vertices and connections, and because the mechanical structure is less rigid. A 3 connected mesh allows the fast meshing of a cloud of points with interactive guidance of calculations. We are concerned by the compression of shape information in the context of object deformation and not in terms of file size.

In this paper we start from a control mesh at subdivision level j=0. The control mesh does not have to be 3 connected. It can represent a closed surface of genus 0 or greater than 0, it can present discontinuities (creases, darts, boundaries) and holes.

Conference poster proceedings ISBN 80-86943-03-8
WSCG'2006, January 30-February 3, 2006
Plzen, Czech Republic.

Internally our software handles only triangular meshes, but we have developed an algorithm that can convert a subdivided triangular mesh into a 3 connected mesh, with fewer vertices and faces.

2. MESH SUBDIVISION BY INSERTION OF A VERTEX ON A FACE

The $\sqrt{3}$ subdivision pertains to this subdivision family. A vertex is inserted in each face. Then new faces are created joining the new vertices to the initial vertices and to the new vertices in immediate neighborhood.

Doing this, a vertex shares 6 faces. From an even level j, we obtain an odd level j+1. It is an "odd subdivision". From the odd level j+1, we obtain an even level j+2. It is an even subdivision. These two successive applications of the algorithm on a triangle, gives 9 triangles. Two applications of the butterfly method, by insertion of points on edge would create 16 triangles [DLG90]. For our purpose the $\sqrt{3}$ subdivision method is better than the butterfly method, because it creates less faces between two even steps. We have more control on the final number of faces. Without creases, darts or boundaries, we use two $\sqrt{3}$ subdivision methods: Approximation of Kobbelt [K00] or Interpolation of Labsik-Greiner [LG00]. We developed new

algorithms in order to compute the vertices near discontinuities.

First of all, we have to choose a mask for the calculus. It defines which vertices of the mesh are going to be used to compute the position of the inserted vertex. Of course, the bigger the mask is, the smoother the limit surface will be, i.e. the limit surface will be Cn, with a higher n. It is confirmed by the calculation of the coefficients for different masks.

We use the method that J.P.Gourret used in [G87] to compute the potential and the electric field near surfaces of reversed biased PN junctions.

3. TRANSFORMATION INTO 3-CONNECTED MESH

To transform a mesh into a 3 connected mesh, we keep only the vertices that have been generated at the last subdivision, and we create a face for each (removed) vertex. The face is made up of all vertices connected to the removed vertex.

4. CREASES

Subdivision software would be useless without creases handling. A crease is a serie of vertices that model a natural discontinuity of the surface. So we cannot take a mask crossing a crease. The article [HDD94] references 3 kinds of natural discontinuities. Crease : it is a deformable crease (it can be smoothed). Corner : it is a non deformable crease. Dart : it is the extremity of a crease or corner. Our software will handle open creases (with 2 darts) or closed creases (cyclical). For example in the article we find the case shown on Fig.1a. The resulting mesh after our 4 subdivisions is shown on figure 1 b&c.

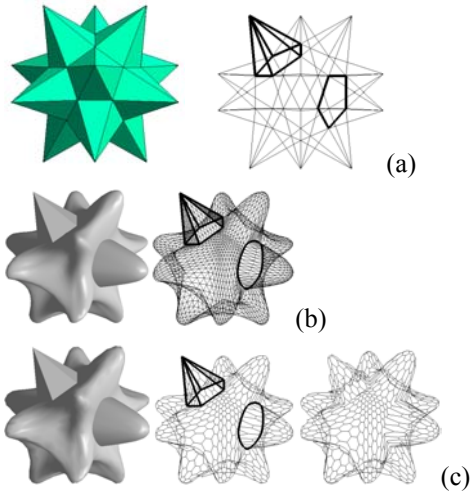


Figure 1: kepler (a) control mesh, subdivision with creases (b) triangular, (c) 3-connected

5. BOUNDARIES

Boundaries allow us to handle open meshes as the one shown in figure 2.

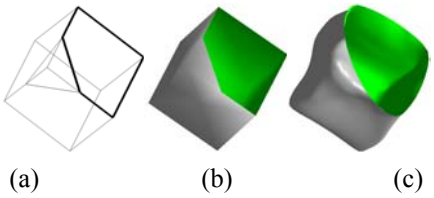


Figure 2: a&b) open mesh (c) after 4 subdivisions

A boundary is a limit, always closed. We can handle as many boundaries as we want, but they should not cross themselves. The idea we use to handle boundaries is to consider it as a crease and create a temporary vertex for each boundary in order to "close" the mesh, so we can reuse the same algorithm than without boundaries.

6. CONCLUSION AND WORK IN PROGRESS

We create a subdivisor that handles creases corners, darts and boundaries. The successive subdivisions of the impulse response of a n connected vertex are shown on Fig. 3. It can be considered as scale functions to establish wavelet functions in a multiresolution analysis. So we are establishing a compression method based on this multiresolution representation by wavelets taking into account creases, corners, darts and boundaries.

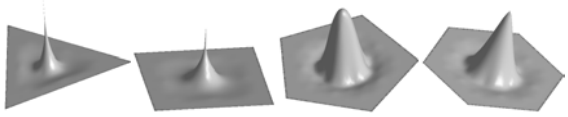


Fig.3: impulse response for a vertex n-connected

7. REFERENCES

[KG04] Khamlichi J., Gourret J.P. « MEFP3C : Un système logiciel pour le maillage évolutif de formes avec pavage par polygones à sommets 3 connexes»., CNRIUT. Tome 1,Univ. Nice, 2004, p.87 et Annexe au tome 1, p.81-88.

[DLG90] Dyn N.,Levin D., Gregory J.A. « A butterfly subdivision scheme for surface interpolation with tension control»., ACM Trans. On Graphics, Vol 9(2), 1990, P.160-169.

[K00] Kobbelt L. « $\sqrt{3}$ *subdivision », Proc. SIGGRAPH,. 2000, P.103-112.*

[LG00] Labsik U., Greiner G. « interpolatory $\sqrt{3}$ *subdivision », proc. EUROGRAPHIC., Vol 19(3), 2000.*

[G87] Gourret J.P.,« Modélisation bidimensionnelle de dispositifs semiconducteurs de puissance mésa et planar»., Rev. Int. De CFAO et d'infographie, Vol 2(2), 1987, p.7-34.

[HDD94] Hoppe H., DeRose T., Duchamp T., Halstead M., Jin H., McDonald J. « Piecewise smooth surface reconstruction», CoProc. ACM SIGGRAPH, 1994, p.295-302.

Fast GPU-based normal map generation for simplified models

Jesús Gumbau
Universidad Jaume I, Spain
jgumbau@sg.uji.es

Carlos González
Universidad Jaume I, Spain
cgonzale@sg.uji.es

Miguel Chover
Universidad Jaume I, Spain
chover@uji.es

ABSTRACT

This paper presents a method for normal map generation in the GPU. These normal maps are generated from a high resolution mesh and can be applied to any simplification of this mesh. This method takes advantage of the fact that there must be a correspondence between the texture coordinates of the low resolution mesh and the ones of the high resolution mesh. The proposed method for normal map generation is a brand-new method, since nowadays this process is being performed through software techniques. Hardware generation greatly reduces time in comparison with present-day solutions. Moreover, it allows for a dynamic modification of the map. There are some restrictions in relation to how texture coordinates must be distributed. However, this approach works perfectly with simplified models where these restrictions are fulfilled. This method makes use of vertex and pixel shaders for the normal map generation.

Keywords: normal map, GPU, shaders, hardware, simplified mesh.

1 INTRODUCTION

The presented method proposes a fast hardware generation of normal maps that uses *vertex* and *pixel shaders*. This idea involves a real-time normal map generation from the high-level object.

The final quality of the resulting normal map is comparable to those obtained from other software-based solutions [1] [3] when applied to multirresolution and simplified models.

Two restrictions have to be accomplished:

- Let $\vec{t_i} = (u_i, v_i) \ \ \forall i \in \{1,2,3\}$ be the texture coordinates for each vertex on a triangle T_j, where T_j is a given triangle of the high resoluction mesh. So, the condition $\bigcap T_j = \emptyset \ \forall T_j \in triangles(Mesh)$ must be fullfiled. However, this requirement is studied in the literature [2][4][5].
- Texture coordinates have to be distributed so that the texture should be correctly applied to both models.

This method works perfectly for terrains and walls, because these objects usually meet the requirements.

2 METHOD

Unlike the present-day normal map generation methods, the presented method generates the maps by *hardware*, by making use of *vertex* and *pixel shaders*.

Posters proceedings ISBN 80-86943-04-6
WSCG'2006, January 30 – February 3, 2006
Plzen, Czech Republic.

Object Space Normal Maps The tasks performed by the *shaders* for normal map generation are:

- The *vertex shader* flattens the image by doing: (x=u, y=v, z=0), where (x,y,z) are the coordinates of the vertex, and (u,v) are its texture coordinates.
- The *pixel shader* generates the normal map by outputting the normal coefficients as RGB components per pixel. For this purpose, it is necessary to convert the normal values into the accepted range by the RGB plane, that is, [0,1].

Tangent Space Normal Maps Moreover, for the normal map generation in tangent space the following extra tasks must be performed:

- The *vertex shader* will receive the tangent space vectors (normal, tangent and binormal) as vertex attributes and will output them so that each fragment will receive these values linearly interpolated.
- The *pixel shader* must transform each normal per pixel into tangent space. The inverse tangent space matrix is already calculated in the *vertex shader* and passed to the *pixel shader*, so transforming each normal to tangent space is as easy as multiply the object space normal by this matrix.

Once calculated, an extra pass must be performed in order to expand texture borders. This is needed to avoid artifacts when filtering is enabled at application time.

3 RESULTS

The presented method has been tested with some 3D models. The obtained times are not comparable with

Polygons of the original model	Time of our method	Time of ATI	Time of nVidia
1696	0.08	16850	16306
7910	0.53	24125	50589
48048	4.74	97704	160975
61644	6.02	129969	179569

Table 1: Table of times in milliseconds of normal map generation

those of present-day *software* methods, since a few milliseconds are taken by the presented method to generate the corresponding normal map.

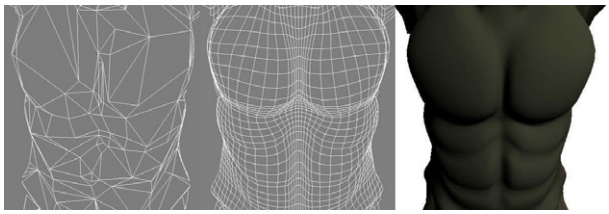


Figure 1: Low and high resolution model meshes of *Tarrasque* (725 and 6117 polygons) with the rendered model

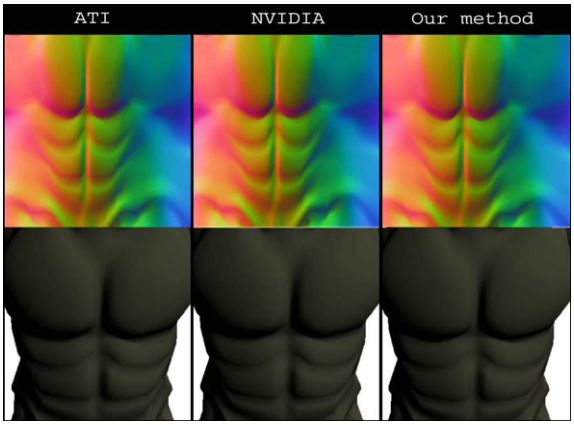


Figure 2: The normal maps in object space of the high resolution model generated and the simplified version with this normal map applied

Measured times using an AMD Athlon64 3500+ with an nVIDIA 6800 can be observed in Table 1.

Figure 1 shows the meshes of both a simplified model and the high resolution model of *Tarrasque*, and the high resolution model rendered.

Figure 2 compares the quality of this method with ray-tracing based methods. The figure shows the generated normal maps and the low resolution model of *Tarrasque* with these normal maps applied, using the methods proposed by ATI, nVidia and our approach, respectively.

The quality of our method is similar to that of the ATI's and nVidia's methods, as seen in the images.

Moreover, the method also works for terrain and wall objects because they usually meet the requirements of this method. Figure 3 and Figure 4 show an example with the *Crater* model.

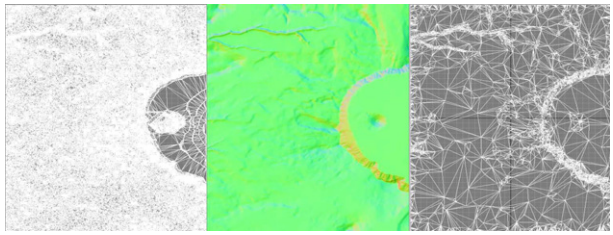


Figure 3: Low and high resolution model meshes of *Crater* and the normal map in object space

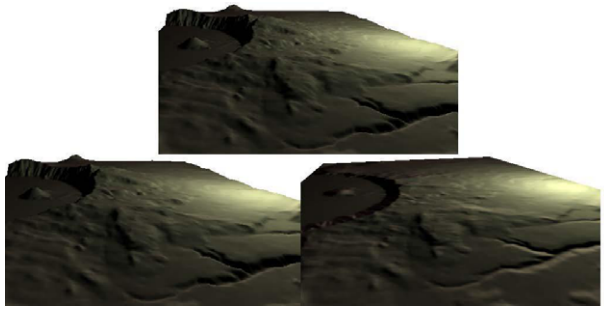


Figure 4: Renders of the high resolution terrain (above), low resolution one with the normal map applied (left bottom) and a plane with the normal map applied (right bottom)

ACKNOWLEDGMENTS

This work has been supported by the Spanish Ministry of Science and Technology (TIN2004-07451-C03-03), the European Union (IST-2-004363) and FEDER funds.

REFERENCES

[1] ATI. Normal Mapper Tool, 2002. *http://www.ati.com/developer/tools.htmlAkeley*

[2] Igarashi, T., Cosgrove, D. Adaptative unwrapping for interactive texture painting. *Symposium on Interactive 3D Graphics 2001*, pp. 209-216

[3] nVidia. nVidia Melody User Guide, 2004. *http://developer.nvidia.com/object/melody_home.html*

[4] Sander, P. V., Snyder, J., Gortler, S. J., Hoppe, H. Texture mapping progressive meshes. *ACM SIGGRAPH 2001*, pp. 409-416

[5] Sloan, P.-P., Weinsteun, D. Brederson, J. Importance driven texture coordinate optimization. *Computer Graphics Forum (Proceedings if Eurographics '98)* 17(3), pp. 97-104

Conceptual design of a programmable geometry generator

Jesús Gumbau
Universidad Jaume I, Spain
jgumbau@sg.uji.es

Miguel Chover
Universidad Jaume I, Spain
chover@uji.es

ABSTRACT

Current real-time graphics architecture lacks a method for procedural geometry generation inside the GPU, so that the limited bus bandwidth doesn't get involved. This document describes the conceptual design of a user-programmable geometry generator unit. This unit is capable of generating new geometry (vertices and indices) by processing a set of input data. This new geometry can be passed through the graphics pipeline to be rendered normally. This is done completely inside the GPU.

Keywords: Pipeline, buffer, GPU, shaders, vertices, fragments.

1 INTRODUCTION

Programmable parts of the present graphics hardware are designed to transform the properties of input primitives (vertices or fragments), through small user programs (also known as "shaders"). However, they are unable to generate new primitives inside the graphics pipeline. This work introduces the conceptual design of a hardware unit capable to generate geometry, inside the GPU, computed from an arbitrary set of input data. This unit will be called *General-Purpose Geometry Generator* (or *GPGG*). It is designed to run in a completely transparent way to the present design of the graphics pipeline, so that the generated geometry can be treated normally by the pipeline.

2 HOW THE GPGG WORKS

The GPGG can be defined as a programmable geometry generation unit designed to work completely inside de GPU. By having this unit inside the GPU, the generated data can be directly sent from the GPGG to the graphics pipeline. Thus, there is a minimal AGP/PCIE bus traffic and thus, no limitations due to the common bus bandwith bottleneck.

The input data that the GPGG can read is defined as a set of generic data that can be stored in any format. It is also possible that the unit doesn't require any kind of input data, so the input data stream is not mandatory. The input data stream is divided in a number of separated input data channels, implemented as different hardware buffers in the GPU. Note that we are talking about general input data, without defining any kind of format for it. This is due to the fact that the GPGG is a general-purpose geometry generator, each buffer can have its own data format.

The processing data unit must have the ability to iterate between input channels, compute the new geometry and generate an output. This should be a user-programmable processor specially designed to work this way. It needs an instruction set generic enough to do any kind of computation from input data and the ability to write into buffers stored in graphics hardware memory: output data channels.

Output data is the result of the geometry generation process. Unlike input data, output data forma must be either vertices or indices. Output data is dumped over memory buffers accessible from the 3D API, so that the result of the generation can be used to feed the graphics pipeline and then the geometry can be rendered. Output data is also divided in different data channels or streams, because the GPGG output data may need to be stored in different buffers: vertex coordinates, texture coordinates, normals, indices or vertex attributes for *vertex programs*.

3 INTEGRATION INTO THE PRESENT ARQUITECTURE

Integration is done by conceiving the geometry generator as a processor separated from the pipeline, binding its input and output channels to buffers stored in graphics hardware. This aproach allows this integration without modifications of the pipeline, just additions.

The graphics pipeline has been designed to transform primitives (render process), not to generate new ones procedurally. The scheme presented in this work respects this idea completelly, separating the *geometry generation stage* (GPGG) and the *geometry representation stage* (graphics pipeline). That makes this integration scheme conceptually cleaner.

Although the general rule is to communicate the GPGG and the pipeline through hardware buffers, the possibility to send automatically the output data to the pipeline is also interesting. This can save a lot of memory for applications that doesn't really need to

Posters proceedings ISBN 80-86943-04-6
WSCG'2006, January 30 – February 3, 2006
Plzen, Czech Republic.

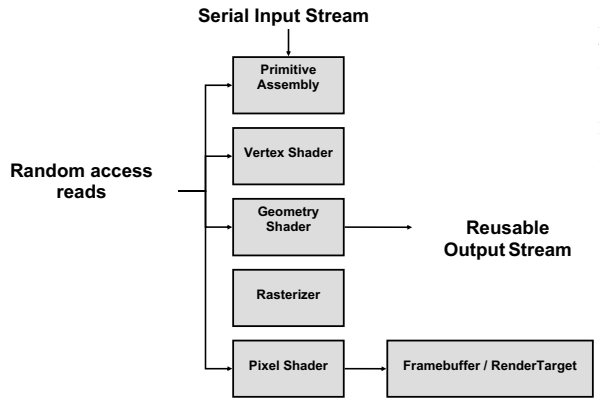


Figure 1: New pipeline proposed by Microsoft

store anything in graphics memory because they need to generate the geometry every time.

To add this capability, an output channel can be configured as a *bridge* to the pipeline, instead if a binding to a memory buffer.

4 GPGG APPLICATIONS

The following is a list of possible real world applications for the GPGG that shows its entire functionality. The GPGG is a perfect tool to calculate the shadow volumes for the *Stencil Shadowing* technique [1] entirely inside the GPU. Every continous LOD technique would benefit from the GPGG as it allows to calculate in real time the triangle list (indices) that define a mesh in an arbitrary level of detail.

Terrain generation [3] could use the GPGG to generate the piece of terrain seen at a given time and a camera position from a heightmap. Displacement mapping coulg be implemented inside the GPU using a similar approach. A surface tesselator [4] could be programmed into a geometry program, setting as input data the control points that describes a bicubic surface. The GPGG could accept as parameters the coeficients of an equation that represents a volume of an object and apply the Marching Cubes method to approximate a polygonal surface.

The unit could be configured to instantiate the geometric primitives for plant generation, by specifying a string, derived from an L-system, as a GPGG input.

5 CONCLUSIONS

Implementing a geometry generator in the graphics hardware has a large amount of benefits, as previously explained (see section 4).

Having the GPGG separated from the pipeline is beneficial in terms of parallelism: while one unit is generating new geometry, the other one can process the already generated geometry.

Even the DirectX approach (figure 1) [2] has multiple *geometry shader* units running in parallel. The same approach could be taken in a hardware implementation of the GPGG by setting up multiple GPGG units running in parallel, similar to the multicore CPU systems do.

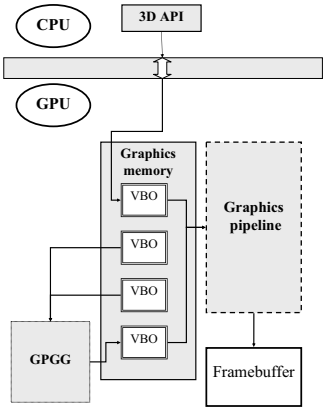


Figure 2: GPGG integration scheme into the current graphics arquitecture.

This work introduces a conceptual design of a hardware geometry generation unit that operates in a completelly transparent way to the current pipeline design.

The changes introduced by Microsoft to implement a geometry generator into the pipeline for DirectX, forces a completelly redesign of the pipeline. In contrast, the GPGG design introduces no changes to the traditional pipeline, only additions to the hardware graphics and the APIs. Moreover, the possibility to access randomly the input data makes it a more intuitive programming model, in contrast to the per-primitive pipelined one proposed by Microsoft.

ACKNOWLEDGMENTS

This work has been supported by the Spanish Ministry of Science and Technology (TIN2004-07451-C03-03), the European Union (IST-2-004363) and FEDER funds.

REFERENCES

[1] Crow, F., *Shadow Algorithms for Computer Graphics*, Computer Graphics, 1977.

[2] Rudolph B., Glassenberg S., *DirectX and Windows Vista*, PDC, 2005.

[3] Losasso, F. and Hoppe, H., *Geometry clipmaps: Terrain rendering using nested regular grids*, Siggraph, 2004.

[4] Sfarti, A., *Bicubic surface rendering*, U.S. patent, #6.563.501.

Extension models of Cone Tree Visualizations to Large scale Knowledge base with Semantic Relations

Daisuke MIZUKOSHI
The 3rd Technical Group Tokyo Business Establishment
Cyber Com Co. Ltd.
Jichodo BLDG 2F 3-3-15 Kaigan
Minato-Ku, Tokyo
108-0022 JAPAN
mizukosi.daisuke@cy-com.co.jp

Yukio HORI
Computing and Communications Center
Kagawa University
Saiwaicho 2-1, Takamatsu-City, Kagawa
761-8521, JAPAN
horiyuki@cc.kagawa-u.ac.jp

Tomonori GOTOH
Department of Information Science
Kanagawa University
Tsuchiya 2946, Hiratsuka-City, Kanagawa
259-1293, JAPAN
gotoh@info.kanagawa-u.ac.jp

ABSTRACT

Cone Tree is an appealing interactive 3D visualization model for hierarchical data structure. In any prior studies, data objects for visualization were constructed by only tree structure, which contained small number of data and nodes. Subject domains in real world for visualization studies have highly complicated relations, which cannot to be expressed in a few nodes and only hierarchical structure. In this paper, we proposed the visualization technique based on cone tree model to apply for a large-scale knowledge base, which has complicated data structure. The EDR Electronic Dictionary as a large-scale knowledge base was used in our study. The visualization system fro EDR was implemented with Java 3D. This paper describes the technique and the implemented system, and discusses some problems on the technique.

Keywords

Cone Trees, visualization, navigation, structural views, lexical knowledge base.

1. INTRODUCTION

Information Science is developed into knowledge science and its importance will increase more and more in recent years. As knowledge data get larger and more complex, they become more difficult to understand its whole structure. The maintainer, who tries to understand knowledge data, reads some semantic relation, and searches the relation and documentation for a confirmation of the conjectures. In developing this prototype tool our primary goal is to help maintainers to easily verify conjectures on the knowledge base with semantic relations.

Interactive visualizations of complex information spaces are designed to improve the ability of users to work with, and navigate through, rich data spaces [Mar97]. Examples include the 3D data displays provided by cone trees [Rob91], the Data Mountain [Rob98], the Perspective Wall[Mac91], and the rich 2D displays of the Information Mural[Jer98]. A fundamental limitation of the work on visualizations is the lack of using few data set and types [Mun98a]. We are unaware of any prior attempt to use large scale data or to use complex data type of cone trees.

In this paper we describe a cone tree visualization models that enables users to visualize large scale knowledge with semantic relations. This system gives the user the ability for the purposes of large scale knowledge analysis.

In this paper we describe a cone tree visualization models that enables users to visualize large scale knowledge with semantic relations. This system gives the user the ability for the purposes of large scale knowledge analysis.

Posters proceedings ISBN 80-86943-04-6
WSCG'2006, January 30-February 3, 2006
Plzen, Czech Republic.

2. VISUALIZATION MODEL

First of all, we describe about EDR electronic dictionary used by our cone tree system [Edr]. The

EDR is developed for the language processing and composed of some large-scale individual dictionaries like the word dictionary etc.

Our cone tree system uses the EDR concept dictionary to display the conceptual system. The broader/narrower relations of the EDR electronic dictionary are huge concept trees which contains more than 400,000 concepts. Each concepts have a unique concept identifier by which they are identified. Lots of concept might have two or more parents concepts because multiple broader concepts.

3. VISUALIZATION EXPERIMENTS

The visualization system based on above mentioned models has been implemented with Java/Java 3D APIs. Java 3D APIs supplies some basic navigation functions by default, i.e. moving angle, rotation of 3D objects, and so on.

First, the coordinates of the apex of the cone is computed and determined. The value of the apex at X axis is equals to it of the rightist node located at X axis, which is located in the same label of the target node.

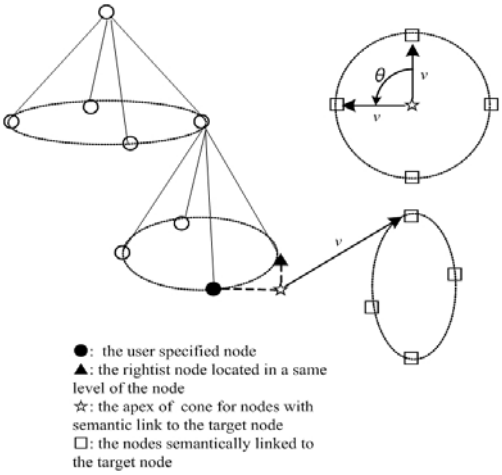


Figure 1. An Image for the relative locations

Figure 2 indicates the example of visualizing the hierarchy tree and the nodes which have semantic relation to the user specified node.

We propose the method of excluding nodes as objects to be visualized, which have no semantic relations. In this method, visibility for semantic relation will be improved, and in addition, hierarchical structure is still maintained. This function as a command has been implemented in our visualization system

4. CONCLUSTION

Cone trees visualizations of hierarchical data structures were first described by Robertson et al [Rob91]. Since then, there have been numerous refinements on the model, but there has been notable applications to only simple tree data structure. This paper provided the visualization technique to complex data structures which gives user to be able to browse and navigate large scale knowledgebase.

In our further work we will investigate the techniques for improvement of operability and visibility for large scale knowledgebase.

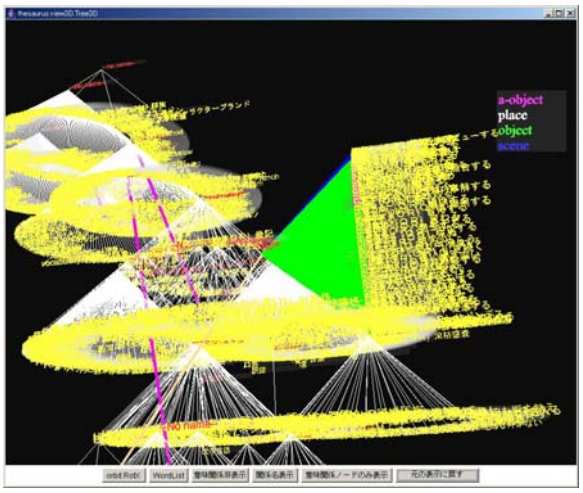


Figure 2. An example for visualization of semantic links between nodes over different trees

5. REFERENCE

[Edr] EDR: http://www2.nict.go.jp/kk/e416/EDR/J_index.html

[Mun98a] Munzner T, Exploring Large Graphs in 3D Hyperbolic Space, IEEE Computer Graphics and Applications, 18(4), pp.18-23, 1998.

[Rob91] G. G. Robertson, J. D. Mackinlay, and S. K. Card., Cone Trees: Animated 3D Visualizations of Hierarchical Information, In Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'91), pp. 189-194. ACM Press, 1991.

[Mac91] J.D. Mackinlay, G.G. Robertson, and S. K. Card, Perspective wall: Detail and context smoothly integrated, In Proceedings of CHI'91 Conference on Human Factors in Computing Systems, pp. 173--179, 1991.

[Jer98] D.F. Jerding and J.T. Stasko, The Information mural: A technique for displaying and navigating large information spaces, IEEE Transactions on Visualization and Computer Graphics, 4(3): 257-271, 1998.

[Mar97] Marti A. Hearst, Chandu Karadi, Cat-a-Cone: an interactive interface for specifying searches and viewing retrieval results using a large category hierarchy, Proceeding of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 246-255, 1997

Image Reconstruction Based on Combination of Wavelet Decomposition, Inpainting and Texture Synthesis

Huawei CHEN
Tokyo Institute of Technology,
2-12-1, O-okayama, Meguro-ku, Tokyo, Japan
chen_hua_wei@yahoo.com

Ichiro Hagiwara
Tokyo Institute of Technology,
2-12-1, O-okayama, Meguro-ku, Tokyo, Japan

ABSTRACT

Digital inpainting provides a means for reconstruction of damaged portions of an image. Although the inpainting basics are straightforward, most inpainting techniques published in the literature are only suitable for remarkable small portion or smooth color image. In order to avoid such shortcomings, we present a new algorithm for digital reconstruction based on combination of wavelet decomposition, Surface-based/PDE-based inpainting and texture synthesis. In which, wavelet transform at first decomposes the image into high frequency and low frequency level parts. Subsequently, CSRBF which is generally used for surface interpolation or PDE-based inpainting is employed for low frequency level and texture synthesis is used for high frequency level. It results in that not only slightly portion but also the common blotched image can be reconstructed with high quality. Especially, our algorithm makes many difficult cases for other methods possible.

Keywords

Inpainting, CSRBF, Wavelet transform, Texture Synthesis

1. INTRODUCTION

Digital photograph has been becoming popular vision device to produce electronic images as the alternative of traditional analog image. In addition to the immediate availability of photos for viewing and/or electronic transfer to an editorial office, digital cameras surpass in various respects such as ease to store and copy without loss of quality for the next upcoming decades. Although these advantages may sound great, one has to consider that a large amount of analog images have to be digitized in order to remain for ever, as well antique artifact or calligraphic works needs to be scanned. Unfortunately, this material often exhibits defects such as scratches or blotches. Even with respect to digital photograph or videos, there are generally many disturbing obstacles or artifacts to be removed, for instance, subtitles, logos, and physical objects such as wires and microphones.

Posters proceedings ISBN 80-86943-04-6
WSCG'2006, January 30-February 3, 2006
Plzen, Czech Republic.

Image reconstruction literatures presented until now can be categorized into two approaches, PDE-based [Bertalmio 00] and Surface-based approach [Nikita 02]. Preserving isophotes is, however desirable in PDE-based approach, never perfectly attained in practice. The main problem is that both isophote estimation and information propagation are subject to numerical diffusion. In contrast to PDE, the surface-based approach is fast and simple to implement. However, this method has no provisions for preserving the isophotes' directions. High-gradient image areas generally are blurred or even not to converge. As another disadvantages, the inpainted region, in general, will be blur or smoothed in unnatural manner as the missed region is considerable large or image gradient is very sharp.

In order to avoid drawbacks mentioned above concerning with two methods, we proposed a new convenient method to automatically repair damaged areas in digitized photographs. The original photo is firstly decomposed into high-frequency and low-frequency parts based on wavelet multi-resolution decomposition. Image inpainting, texture synthesis and noise removal are adopted for low-frequency and high-frequency respectively. The efficiency of proposed reconstruction method is to be clarified by diverse practical implementations.

2. Overview of Our Method

As mentioned in the introduction, blemished photograph includes not only photograph with small spot or scratch which can be considered noise but also that with large damaged area. Image inpainting or reconstruction algorithm is necessary to take above two cases into consideration. Under such requirements, one novel and convenient algorithm combining the texture synthesis and inpainting is proposed as shown in Fig.1:

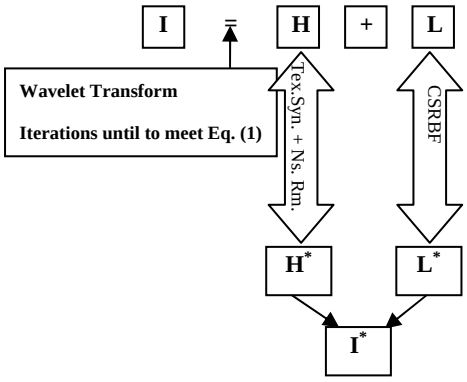


Fig.1 Procedure of proposed method (Tex. Syn.: Texture Synthesis; Ns. Rm.: Noise Removal)

1) The input image *I* is decomposed into a high frequency part *H* and low frequency part *L* by wavelet multi-resolution decomposition.

2) In order to guarantee the property of the inpainted photograph and availability of inpainting method (surface-based or PDE-based), whether or not the gradient Gd_k of *k* pixel to neighboring pixels exceeds the threshold is used to determine the iteration times of decomposition.

$$Gd_k = \frac{\sum_{i:pixels\ around\ k} |C_i - C_k| / 255}{N} \le threshold \qquad (1)$$

where C_i is color value of *i* pixel; *N* is the number of neighboring pixels. The *threshold* is approximately set to be *0.2*.

3) With respect to high frequency parts *H*, multi-resolution texture synthesis is implemented to fill inside the masked areas.

4) With respect to low frequency part *L* in general denoted ***LL***, local *CSRBF* (Compactly Supported Radial Basis Function) developed on our own is used to modify the masked area.

5) Wavelet transform is frequently used to eliminate the random error in original signal. In the duration of image reconstruction, we also employ it to eliminate the small spot considered as noise.

6) Finally, the image *I** will be reconstructed by use of inverse discrete wavelet transform.

3. Experiment Result

Fig. 2 shows the inpainting results for damaged Japanese cherry tree in which the boundary color of damaged area change more remarkably as compared with Lena photograph depicted above. The original Japanese cherry tree image in which the damaged area is approximately identical to Lena in size is shown in Fig. 2(a). Fig.2 (b) and (c) show the reconstructed results on basis of iterative gradient interpolation algorithm proposed by Bertalmio M. and *CSRBF* respectively. Although they can be well adapted to above Lena image, both fail to reconstruct damaged Japanese cherry tree image. As the reason, it can be regarded as that the boundary color surrounding the damaged area changes significantly and the size of damaged area becomes broader. In contrast, the proposed combination reconstruction results in one desirable image which nearly exhibits damaged mark as shown in Fig.2(d). In addition, third-level wavelet transform is employed in the duration of decomposition. That is to say, the shaper boundary of damaged area leads to much more decomposition times in order to make the *CSRBF* available.

(a) Original Japanese cherry tree (b) Bertalmio M.

(c) CSRBF (d) Our Combination Approach

Fig. 2 Image Inpainting for Japanese Cherry Tree

4. REFERENCES

[Bertalmio 00] Bertalmio M., Sapiro G., Caselles V., Ballester C., Image Inpainting, *SIGGRAPH* 2000, p.417-424.

[Nikita 02] Kojekine N., Hagiwara I., and Savchenko V., Software Tools Using CSRBFs for Processing Scattered Data, Computer & Graphics, pp. 309-317, 2002.

Sectional Discrete Curvature Estimation Based on the Parabola

Hyoungseok Kim[1] and Hosook Kim[2]

[1] Dept. of Multimedia Engineering, Dongeui University, KOREA
[2] Division of Internet Information & Multimedia, Dongeui Institute of Technology, KOREA

Abstract

The local geometric properties such as curvatures and normal vectors play important roles in analyzing the local shape of objects. The result of the geometric operations such as mesh simplification and mesh smoothing is dependent on how to compute the curvature of vertices, because there is no its exact definition in meshes. In this paper, we indicate the fatal error in computing discrete sectional-curvatures by the previous discrete curvature estimations. Moreover, we present a new discrete sectional-curvature estimation to overcome the error, which is based on the parabolic interpolation and the geometric properties of Bezier curve.

Keywords: Discrete Curvature and Parabolic Interpolation

1. Introduction

The problem of estimating the geometric properties such as normal vectors and curvatures in triangular meshes plays important role in many applications such as surface segmentation and anisotropic remeshing. A lot of efforts have been devoted to this problem, but there is no consensus on the most appropriate way [1,3,4,5,8,9]. Popular methods typically consider some definition of curvature that can be extended to the polyhedral setting. Taubin presented a method to estimate the tensor of curvature of a surface at vertices of a mesh [6]. Watanabe proposed a simple method of estimating the principal curvatures of a discrete surface [7]. Meyer et. al proposed a discrete analog of the Laplace-Beltrami operator to estimate the discrete curvature[2].

Most of these methods compute directly the sectional curvatures for each adjacent edge of a vertex. They assume that the normal curve interpolates both the given vertex and an adjacent vertex and the curve is represented by Taylor series. However, they make the same mistake that they adopt the distance between the given vertex and its adjacent neighbor vertex as the parameter of the series. There are several polygons of different interior angles, all of which are circumscribed by circles of the same radius. The discrete curvatures of all vertices estimated by those methods are the same as that of the circle although they have different interior angle. It is quite alien to universal concepts.

2. Parabola-Based Discrete Curvature

We adopt a quadratic Bezier curve as an interpolating curve. Let A, B, C be three consecutive vertices. The general form of the quadratic Bezier curve satisfying P(1/2) = B is as follows:

$P(t) = A\,B_0^2(t) + (4B-A-C)/2\,B_1^2(t) + C\,B_2^2(t)$,

where $B_i^n(t) = n!/((n-i)!\,i!)\,(1-t)^{(n-i)}\,t^i$ are the Bernstein polynomials of degree n. The curvature of P(t) at t=1/2 is

$$\kappa_P(1/2) = \| P''(1/2) \times P'(1/2) \| / \|P'(1/2)\|^3 = \| 4(A-2B+C) \times (C-A) \| / \|C-A\|^3.$$

Hence, we can define a new Parabola-based discrete curvature of the given vertex B as follows:

$$\kappa_P(B) \equiv \|4(A-2B+C)\times(C-A)\|/\|C-A\|^3.$$

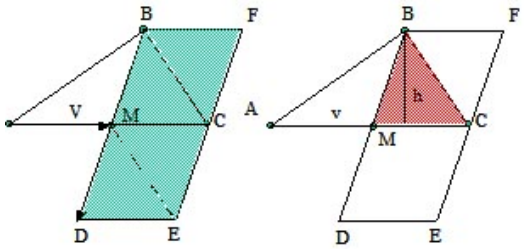


Figure 1. The geometric property

First of all, we find out the geometric properties of the P-discrete curvature formula. Let V=(C-A)/2 and G = A-2B+C. The P-discrete curvature formula is

$$\kappa_P(B) = \| 4G \times 2V \| / \| 2V \|^3 = \| G \times V \| / \|V\|^3 = (\|G\|\,\|V\| \sin\theta) / \|V\|^3, \quad (1)$$

where θ is the in-between angle of the vectors G and V. The numerator of Equation (1) is the area of the parallelogram BDEF and is four times as much as the area of the triangle BCF as shown in Figure 1. Therefore, the P-discrete curvature formula is $\kappa_P(B) = 2h/v^2$, where, h and v are the height and the width of the triangle BCF, respectively.

3. Experimental Results

In order to verify the propriety of the P-discrete curvature, we regularly sample n points on a circle of radius 1 and compute their P-discrete curvature. Let $p_i=(\cos((2\pi\ i)/n), \sin((2\pi\ i)/n))$, i = 0, … , n-1, be the vertices of a n-gon on the circle. By trigonometry, we can compute the values of v and h as follows:

$$v = r \sin((2\pi\ i)/n), \quad h = r(1 - \cos((2\pi\ i)/n)).$$

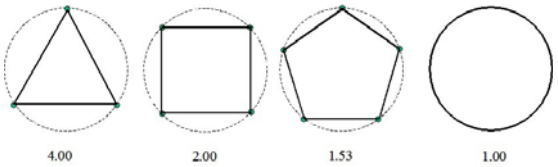


Figure 2. Polygons with the different p-curvature values

Therefore, as the number of sampling points increases to the infinity, the value of curvature at a vertex of the n-gon becomes that of the circle.

$$\lim_{n\to\infty} (2(1-\cos((2\pi i)/n))/r\sin^2((2\pi i)/n) = 1/r.$$

Figure 2 shows the several polygons on a circle of radius 1 and their P-discrete curvature values. The result is an excellent contrast to that of circular based (C-type) discrete curvature estimation. That estimation wishes that the curvature at the sampled vertices may become that of a circle. The method puts emphasis on the point of view that the vertices are on a circle. However, it goes against the concept of curvatures. It loses the information on the local shape. On the other hand, our method recognizes the vertices of polygons to have a sharper angle, not to be on a circle. That is, the P-discrete curvature of vertices of a triangle is 4 and that of rectangle is 2.0 (see Figure 2). More the number of vertices increases to the infinity, less the curvature value decreases to 1. Table 1 shows the differences of C-type discrete curvature and P-type discrete curvature

4. Conclusion

The analysis on the local properties of 3D meshes plays an important role in the applications such as morphing, simplification, smoothing. In special, the curvature at a point on a surface may represent the shape of its neighborhood. However, there is an exact definition of the curvature at a vertex. So, one has to approximate the value as a discrete curvature. The common previous methods compute directly the sectional curvatures for each one-ring neighbor, and then derive the Gaussian curvature and the mean curvature using the sectional curvatures. All of them utilize the circle-based discrete curvature to compute the sectional curvature. In this paper, we find out a fatal mistake and propose the parabola-based discrete curvature estimation to resolve the problem. Our method may be the basis of normal vector estimation and segmentation of meshes.

Table 1. P-type curvature vs. C-type curvature

	C-type Discrete urvature	P-type Discrete Curvature
Formula	$\kappa_C(B) = (2\ N\cdot BA) / \|\| AB\|\|^2$	$\kappa_{SP}(B) = (2\ N\cdot BA) /\|\| BA - (N\cdot BA)N\|\|^2$
Parameter	Distance	Horizontal Distance
Range	$\kappa_C(B) \le 2$ if $\|\|AB\| \ge 1$	$\kappa_{SP}(B) < \infty$
Trajectory	circle	parabola
Magnitude	$\kappa_C(B) \le \kappa_{SP}(B)$	

Acknowledgement

This work was supported by the Post-doctoral Fellowship Program of Korea Science & Engineering Foundation(KOSEF)

References

1. Sylvain Petitjean, " A Survey of Methods for Recovering Quadrics in Triangle Meshes", ACM Computing Surveys, Vol. 34, No. 2, pp 211—262 (2002).
2. Mark Meyer, Mathieu Desbrun, Peter Schroder, and Alan H. Barr, "Discrete Differential Geometry Operators for Triangulated 2-Manifolds", Proc of VisMath, (2002)
3. J. Goldfeather and V. Interrante, " A novel cubic-order algorithm for approximating principal direction vectors", ACM Transaction on Graphics, Vol . 23, No. 1, pp 45—63 (2004)
4. Holger Theisel, Christian Rossl, Rhaleb Zayer, and Hans-Peter Seidel, "Normal Based Estimation of the Curvature Tensor for Triangular Meshes", Proc. of Pacific Graphics'04, pp 288—297 (2004)
5. Tatiana Surazhsky, Evgeny Magid, Octavian Soldea, Gershon Elber, and Ehud Rivlin, " A Comparison of Gaussian and Mean Curvatures Estimation Methods on Triangular Meshes", Proc. of 2003 IEEE International Conference on Robotics and Automation, (2003)
6. Gabriel Taubin, "Estimating the Tensor of Curvature of a Surface from Polyhedral Approximation", Proc. of the Fifth International Conference on Computer Vision, (1995)
7. Kouki Watanabe and Alexander G. Belyaev, "Detection of Salient Curvatures Features on Polygonal Surfaces", Computer Graphics Forum, Vol. 20, No. 3, (2001)
8. David Cohen-Steiner and Jean-Marie Morvan, "Restricted Delaunay Triangulations and Normal Cycle", ACM Symposium of Computational Geometry, (2003)
9. M. Garland and P. S. Heckbert, "Surface simplification using quadratic error metrics", SIGGRAPH '97, pp 209—216 (1997)

A proposal from Information Visualization and Human Computer Interaction point of view to the Design of Industrial Interfaces

Larrea, Martín L. Martig, Sergio R. Castro, Silvia M.
Departamento de Ciencias e Ingeniería de la Computación
VyGLab - Laboratorio de Investigación en Visualización y Computación Gráfica.
Universidad Nacional del Sur
Avenida Alem 1253
Argentina, CP 8000, Bahía Blanca, Buenos Aires
mll@cs.uns.edu.ar srm@cs.uns.edu.ar smc@cs.uns.edu.ar

ABSTRACT

In this paper we show how the design of the interfaces present in a control room can benefit from the combination of concepts and techniques from Information Visualization and Human Computer Interaction (HCI). The process monitoring GUIs must present all the necessary information to control and change the state of the plant in a safety and productive way. Every person inside a control room must handle a high information volume and the exploration and analysis of this huge data volume has become increasingly difficult. This difficulty can lead to risk not only the production but also human lives. Information Visualization and HCI can help to deal with the flood of information.

Keywords

Industrial Graphics Interfaces, Process monitoring, Information visualization, Focus+Context Visualization.

1 INTRODUCTION

The design of the interfaces present in a control room can be tackled from two points of view, showing the states of each process in an effective way and providing all the necessary interaction to control theses processes. Process monitoring tools base their representation in a mimic visualization. This technique creates visual images that exactly match the process they represent. However, these visual representations may often do not fit the actual size of the display, forcing the user to see only a portion of the process. Because of this limitation, the user can only handle a part of the process information while losing the remain of it. The problem of exploring large data sets has been studied over the last decades within two major areas as Information Visualization [Kei02] and Human Computer Interaction [Shn98]. In this paper we show how the design of the interfaces present in a control room can benefit from the combination of concepts and techniques from Information Visualization and Human Computer Interaction (HCI). We apply the results obtains from these areas to create an effective visualization tool that reduces the user cognitive workload.

Posters proceedings ISBN 80-86943-04-6
WSCG'2006, January 30-February 3, 2006
Plzen, Czech Republic.

2 INDUSTRIAL PLANT VISUALIZATION

Each industrial plant can be seen as a single process that receive some input and produces an output. This view may not be sufficient to control the state of the plant and a more detailed representation might be needed. If we replace this single process with all the main ones, we achieve an equivalent view but with more information. This decomposition of processes can be repeated for each process and continue until we reach a level where no new process can be added.

An industrial plant is a set of processes interconnected. Theses links between them create a graph where each node represent a process and the edges are the connections between them. Hence it is correct to say that the problem of visualizing industrial plant processes and its states is equivalent to visualize a graph and all information relative to each node and edge.

Today's techniques divide the plant into sectors and allow the user to navigate them. These techniques don't allow the user to combine different levels of detail at the same time in the same view and have a lack of context on the user position. Theses are the problems that we seek to solve combining the work done in Information Visualization and HCI with the develop of industrial control interfaces. Our main objective is to obtain a technique that present all the necessary information to control and change the state of the plant in a safety and productive way.

3 A NEW PROPOSAL

We propose to combine different level of the multiresolution structure in the same view. The user will be able to increase or decrease the level of detail in a specific area, both semantic and graphically. To avoid the situation where the entire graph does not fit in the display's area, we define a constant value **max** that represent the maximum number of nodes allowed on a view. When the number of nodes exceeds the given value **max**, automatic implosions are produced on the periphery of the graph. The main purpose of this is to reduce the number of elements in the display with a minimum disorientation of the user.

A change in the level of detail is not only a graphic modification but also a semantic one. When the user increases the level of detail on the focus, we provide more information and possibly, but not necessarily, a different graphic representation. The same situation occurs when the user decrease the level of detail.

When the user focuses on an element of the graph, we use the fisheye view transformation [Sar94] to modify the current view. By doing this, the element selected gain focus and the rest of the graph becomes context.
Since fisheye view applies a geometric transformation, the entire graph is in the view at all time. And while the focus is at one level of detail, the context may present a different one, or even more than one.

The user will be able to specify, for each node, which are the ones he/she wish to see when the node change to an emergency state. This set of elements is called an **emergency graph**. A node with an unknown alarm and its emergency graph will never be hidden from the user automatically (figure 1a-1b). Finally, the user has the option to lock a node so that it never implotes.

4 CONCLUSIONS AND FUTURE WORK

It is possible to transfer the result from Information Visualization and HCI on large data set exploration to the industrial plan control room. By doing this we can apply techniques specially designed for this purpose and that have proved to be successful.

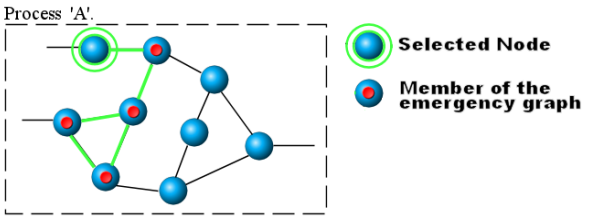


Figure 1a. The nodes with the red dot belong to the emergency graph of the selected node.

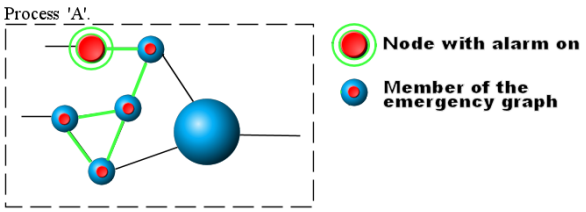


Figure 1b. Implosion of the nodes that are not in the emergency graph.

Our proposal for the visualization includes: Multiresolution graphs that represents the industrial plant and the emergency graph. Geometric zoom, Semantic zoom and the use of Fisheye View to achieve Focus + Context.

Unfortunately, there has been no testing of the usability of this visualization. Much more study of this area is sorely needed. In addition, there needs to be a study of how other transformations techniques can applied to design of the interfaces present in a control room. Current work has been limited to a few interactions between the user and the visualization. In order to recognize the true potential of this technique real interaction must be consider under our proposal.
Future work will involve the develop of a prototype with a simulation of an industrial plant and testing this prototype with real users.

5 ACKNOWLEDGMENTS

This work was partially supported by a grant from the Science and Technology Department of the Universidad Nacional del Sur (SECyT, UNS).

6 REFERENCES

[Kei02] Daniel A. Keim, "Information Visualization and Visual Data Mining". IEEE Transactions on Visualization and Computer Graphics, Vol. 7, No. 1, January-March 2002.

[Sar94] Sarkar, Manojit and Brown, H., Marc, "Graphical FishEye views". Comunication of the ACM. December 1994/Vol 37, No. 12.

[Shn98] Shneiderman, B., Designing the User Interface, Addison-Wesley Publishing Company, 1998.

A Level-based Geometric Representation for the Real-time Simulation of NC Machining Processes

Aitor Moreno[(1)] Carlos Toro[(1)] Iosu Arizkuren[(1)] Álvaro Segura[(1)] Jorge Posada[(1)]
{amoreno, ctoro, iarizkuren, asegura, jposada}@vicomtech.es
Marcelino Novo [(2)] Juanjo Falcón [(3)] Nieves Alcaín [(4)]
mnovo@aotek.es jjfalcon@somesi.com nalcain@alecop.es

[(1)] **VICOMTech.** Paseo Mikeletegui 57. 20009 San Sebastián (Spain)
[(2)] **FAGOR AUTOMATION S. Coop.** Bº San Andrés, 19 - Apdo.144 - 20500 Mondragón (Spain)
[(3)] **SOME Sistemas Informáticos S.L.** Avda. Navarra s/n (oficina 10). 20500 Mondragón (Spain)
[(4)] **Alecop, S. Coop.** Loramendi, 11, Apto. 81 - 20500 Mondragón (Spain)

ABSTRACT

In this paper a level-based geometric representation for a real-time material removal simulator is presented. It will be embedded into a commercial NC machine. The representation and its accompanying architecture have been used for the generation of an interactive simulation of a part being machined, taking the NC machine feedback as input. The simulator complies with the following restrictions: *i*) The virtual simulation and real machining process must be synchronized; *ii*) the internal representation of the objects must be as accurate as possible.

Keywords

NC-Machining, Verification, Material Removal, Solid Representation, Simulator, Interactive Rendering.

1. INTRODUCTION AND PREVIOUS WORK

Numerical Control (NC) machining simulation using Computer Graphics techniques is a widely extended research topic.

Traditional approaches do not store geometrical information during the simulation; they simply modify the drawing screen using an image-based approach. This technique is the basis for several commercial systems for NC machining simulation.

Posters proceedings ISBN 80-86943-04-6
WSCG'2006, January 30-February 3, 2006
Plzen, Czech Republic.

In the other side, there are solutions that store the intermediate result in the computer's memory, storing an internal 3D geometric representation of the object that is changed dynamically during the simulation, for example, B-Rep, CSG and Hierarchical Space Decomposition.

A straightforward implementation with these kind of methods is very time consuming. To cope with this problem, the approximation of the exact geometry, and the partitioning of the object in suitable regions has been proposed by several authors [Jerard89].

In this paper, a level-based solid representation is proposed as the geometrical kernel of a NC material removal simulator.

2. PROPOSED APPROACH

The chosen geometrical representation for the manufactured objects is a level-based representation, consisting of a set of parallel levels to represent the 3D object. Each level is a set of non-intersecting and coplanar polygons with at least three vertices.

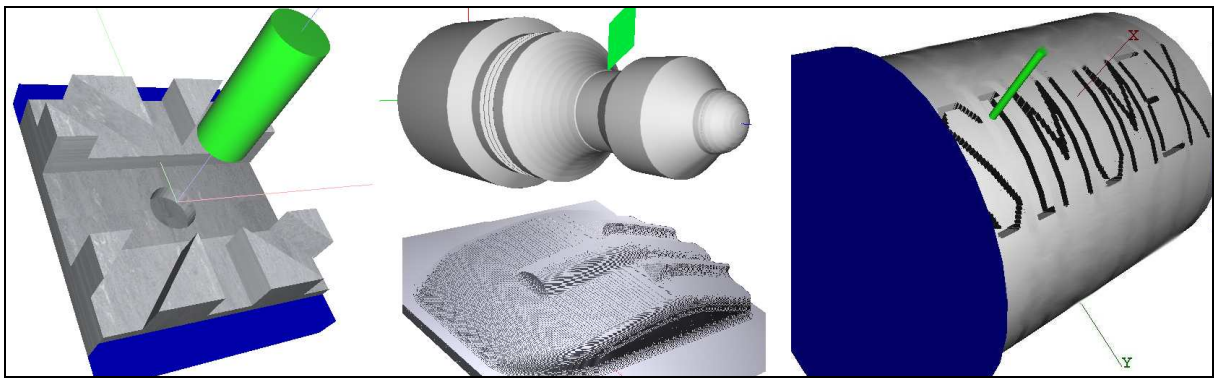


Figure 1: Some simulation results. From left to right: 2D milling, 2.5D lathe, 3D milling and C-axis lathe

The distance between the levels defines the maximum detail that can be perceived. This representation will be called Level-Based Representation (LBR). Thus, LBR object is the object represented using the LBR representation (see figure 2).

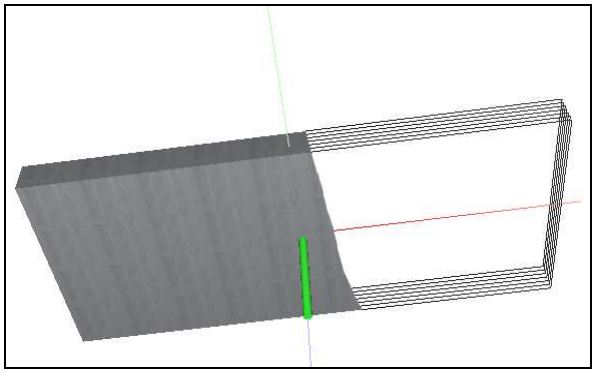


Figure 2: A LBR parallelepiped is shown in solid mode (left) and its approximation in levels (right).

The LBR is a direct way to perform boolean operations between LBR objects: a 3D boolean operation is simplified in a set of 2D boolean operations between two 2D polygons. This approach is a well reviewed research topic [Vatti92] .

As the simulation goes on, the complexity of the manufactured piece increases, which makes the number of points and contours to grow as well. In order to limit the number of points and contours that would increase the boolean operation time, a high-level partitioning system is added to the object definition.

This spatial partitioning decomposes the objects into a set of smaller regions, each of them being a set of parallel levels, as described previously.

The overall computational cost is reduced since the classical boolean operation between geometrical objects is $O(n^4)$ [Pourt01]. This approach reduces the computational cost to $O(n*m)$, where n is the number of levels and m the number of contours per level.

3. RESULTS

We have developed a simulation system embedded in a NC machine using the proposed approach as the internal representation. Taking the feedback of a NC machine as the input for the simulator, the results show that a synchronized simulation is achieved with small and medium NC programs, obtaining an interactive visualization. The tests have been made on an AMD K6 1500 processor with a GeForce 2 graphic card with 32Mb and a 800 x 600 resolution. The NC machine is a 8070 CNC from Fagor Automation (www.fagorautomation.com).

4. CONCLUSIONS

The paper presents a level-based geometric representation as the kernel of a material removal simulator for real time execution when embedded in a real NC machine (see figure 1).

The results have confirmed that a proper real-time simulation is achieved with interactive rates and a high image quality taking into account that the simulator takes its input directly from the NC machine, with no knowledge of the high level NC programming.

5. REFERENCES

[Jerard89] Jerard R. B. R. L. Drysdale, “Methods for detecting errors in sculptured surface machining”, IEEE Computer Graphics & Applications. Jan. 1989, pp. 26 -39.

[Pourt01] K.Poutrain, M. Contensin, “Dual B-Rep-CSG collision detection for general polyhedra”, Computer Graphics and Applications, 2001, Proceedings. Ninth Pacific Conference on 16-18 Oct. 2001 pp. 124 - 133.

[Vatti92] B.R. Vatti, "A Generic Solution to Polygon Clipping", Commun ACM 35 1992, pp. 56 – 63.

Combined filtering and key-frame reduction of motion capture data with application to 3DTV *

Onur Önder
Bilkent University
Computer Engineering Department
Bilkent, Ankara, Turkey
oonder@cs.bilkent.edu.tr

Çiğdem Erdem, Tanju Erdem
Momentum DMT
TUBITAK-MAM TEKSEB
Gebze, Kocaeli, Turkey
cigdem.erdem@momentum-dmt.com
terdem@momentum-dmt.com

Uğur Güdükbay, Bülent Özgüç
Bilkent University
Computer Engineering Department
Bilkent, Ankara, Turkey
gudukbay@cs.bilkent.edu.tr
ozguc@bilkent.edu.tr

ABSTRACT

A new method for combined filtering and key-frame reduction of motion capture data is proposed. Filtering of motion capture data is necessary to eliminate any jitter introduced by a motion capture system. Key-frame reduction, on the other hand, allows animators to easily edit motion data by representing animation curves with a significantly smaller number of key frames. The proposed technique achieves key frame reduction and jitter removal simultaneously by fitting a Hermite curve to motion capture data using dynamic programming.

Keywords
motion capture, key-frame reduction, curve fitting.

1. INTRODUCTION

Human motion capture is an important area of research with applications in diverse fields such as art, entertainment and education [Nguyen05]. It is also regarded as one of the enabling technologies for the ongoing 3DTV development [3DTVNoE]. Although human motion capture systems vary in nature, such as optical, magnetic, mechanic or hybrid systems, the captured motion data in general need to be filtered and edited before they can be used on a human model [Jun00a]. Filtering is necessary in order to eliminate any jitter that may be introduced by the motion capture acquisition system. It is also desired to represent the filtered motion data with a smaller number of samples (key frames) than those that are acquired. Animators can then modify the animation curves more effectively by editing the attributes of a much smaller number of key frames. Key frame reduction requires fitting a spline curve to motion data.

In this paper, we propose an algorithm for fitting a Hermite curve [Foley95] to motion capture data by means of dynamic programming [Amini90]. Hermite curves are easily controlled via the position and tangent vectors at key frames and are generally used by animators for motion editing. The proposed dynamic programming based curve fitting technique achieves combined filtering and key-frame reduction of motion capture data.

Posters proceedings ISBN 80-86943-04-6
WSCG'2006, January 30-February 3, 2006
Plzen, Czech Republic.

*This work was supported by EC FP6 Grant 511568 with the acronym 3DTV, and by TUBITAK TIDEB under Grant 3050135.

2. CURVE REPRESENTATION

A segment of a Hermite curve $v(s)$ between the control points i and $i+1$ is given by (see Figure 1)

$$v(s) = \begin{bmatrix}(s-i)^3 & (s-i)^2 & (s-i) & 1\end{bmatrix} \cdot \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} p_i \\ p_{i+1} \\ r_i \\ r_{i+1} \end{bmatrix}$$

where p_k represent the positions and r_k represent

the tangents at key frame k; they will be called the control vectors of a Hermite curve.

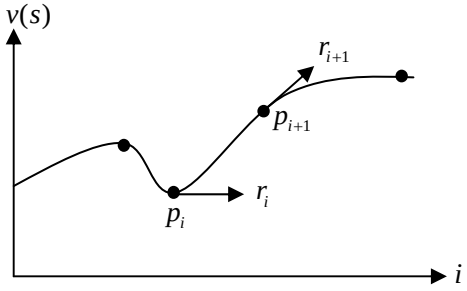


Figure 1. Control points and tangent vectors define the shape of a Hermite curve.

It will be necessary to calculate the value of a Hermite curve between consecutive control points during the curve fitting algorithm presented in Section 3. To this end, a Hermite curve segment is first converted to a Bezier curve [Foley95]. Once a curve segment is converted to a Bezier curve, it becomes straightforward to sample the curve segment at any accuracy. This is because a Bezier curve can be recursively divided in half where each half is also a Bezier curve. The equations for obtaining the Bezier control points from Hermite control points and tangents and the process of recursively dividing a Bezier curve can be found in [Foley95].

3. CURVE FITTING ALGORITHM

The goal of the proposed curve fitting algorithm is to minimize the mean squared error (MSE) between the motion capture data and its curve representation. That is, we would like to find the position and tangent vectors of a Hermite curve so that the overall curve fits the motion data in the minimum MSE sense. Dynamic programming is employed so that all possible combinations of control vectors are taken into consideration by only computing MSE values for individual segments. The proposed algorithm is given as follows:

1. Find initial estimates of p_i, p_{i+1}, r_i, and r_{i+1}, for all i by placing key frames at those locations where the first and second derivatives are zero on the motion capture data. Hence the number and estimated locations of key frames are determined at this step.
2. Determine search spaces around the initial estimates for all vectors p_i, p_{i+1}, r_i, and r_{i+1}. Note that each vector has x- and y- components. In addition to the original component value, use values that are obtained by adding and subtracting a predetermined step to/from the original value. Hence, each search space will have 3 values, and there will be 6561 combinations for each curve segment.
3. Compute the MSE for each combination of p_i, p_{i+1}, r_i, and r_{i+1} in the search space. In order to calculate the value of the Hermite curve at the same frame locations as the captured data, we subdivide every Hermite curve segment based on a corresponding Bezier representation (see Section 2). Save the best values of p_i and r_i for each combination of p_{i+1} and r_{i+1}.
4. When the last segment of the curve is reached, find the total MSE for each combination of p_{i+1} and r_{i+1}. To calculate the total MSE, traverse the curve backwards and use the best (minimum) previous values obtained in Step 3.
5. Pick the overall path of control vectors that results in the minimum MSE in Step 4.
6. Assume that the values p_i, p_{i+1}, r_i, r_{i+1}, for all i, obtained in Step 5 are the updated initial values and repeat Steps 2-6 until a predetermined accuracy in x- and y- components is reached.

4. CONCLUSION

A new algorithm is presented that employs dynamic programming to fit a Hermite curve to motion capture data in the MSE sense. The performance of the proposed technique will be demonstrated on actual motion capture data. Both subjective and objective evaluations of the results will be presented.

5. REFERENCES

[Amini90] Amini, A.A., Weymouth, T.E., Jain, R. C., *Using dynamic programming for solving variational problems in vision*, IEEE Trans. Pattern Anal. Mach. Intel., vol. 12, pp. 855-866, Sept 1990

[Foley95] Foley, J.D., van Dam, A., Feiner, S.K., Hughes, J.F., *Computer Graphics: Principles and Practice in C*, 2nd ed., Addison-Wesley, 1995

[Jun00a] Jung, M., Fischer, R., Gleicher, M., Thingvold, J., *Motion Capture and Editing: Bridging Principles and Practices*, 2000.

[Nguyen05] Ta Huynh Duy Nguyen, Tran Cong Thien Qui, Ke Xu, Cheok, A.D., Sze Lee Teo, ZhiYing Zhou, Mallawaarachchi, A., Shang Ping Lee, Wei Liu, Hui Siang Teo, Le Nam Thang, Yu Li, Kato, H., *Real-Time 3D Human Capture System for Mixed-Reality Art and Entertainment*, IEEE Trans. Visualization and Computer Graphics, Dec. 2005

[3DTVNoE] www.3dtv-research.org

The creation of simple plants by the help of Python script in Blender

Pavel Pokorný
Faculty of Technology in Zlin
Thomas Bata University
Mostni 5139, 760 01 Zlin,
Czech Republic
pokorny@ft.utb.cz

ABSTRACT

Blender is the open source software for 3D modeling, animation, rendering, post-production, interactive creation and playback. This software is fully free (with source code under GNU GPL) to use and distribute within any educational, professional or commercial environment. We can use some modeling methods implemented in Blender or Python scripts for the creation of our artwork. Blender has implemented an embedded Python interpreter, which allows it to run scripts written in that language. These scripts can use the Blender Python API to access the program's internals and greatly expand Blender's functions. In this paper, a simple Python script for the plants generation has been described and parameters and a random factor have been set.

Keywords
Programming, Python script, Blender

1. INTRODUCTION

Blender is an integrated suite of tools enabling the creation of and replay of linear and real-time, interactive 3D content. It offers full functionality for modeling, rendering, animation, post-production, game creation and playback with the singular benefits of crossplatform operability and a download file size of less than 4.8 MB. [Ble05a] [Ble05b]

Python is an interpreted, interactive, object-oriented programming language. Python combines remarkable power with very clear syntax. It has modules, classes, exceptions, very high level of dynamic data types, and dynamic typing. There are interfaces to many system calls and libraries, as well as to various windowing systems (X11, Motif, Tk, Mac, MFC). New built-in modules are easily written in C or C++. Python is also usable as an extension language for applications that need a programmable interface.

Posters proceedings ISBN 80-86943-04-6
WSCG'2006, January 30-February 3, 2006
Plzen, Czech Republic.

2. ABOUT BLENDER

The user communicates with Blender via the keyboard and the mouse, the program gives feedback via the screen and its windowing system (Figure 1). Blender's interface allows using three mouse buttons and a wide range of hotkeys.

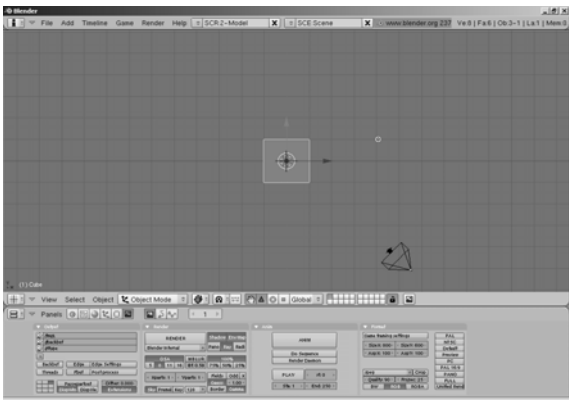


Figure 1. The Blender scene.

The 3D viewport window shows the environment of the current scene in the default viewing direction, but we can change it (Blender offers us to work in three-dimensional or two-dimensional space).

When we create a scene in Blender, we start with a few basic elements that include simple cube object, a camera and a light. We can add more cameras and more lights, but this scene is ready for rendering too.

3. BLENDER SETTING FOR THE PYTHON

Blender supports more types of windows. Each window in Blender has a specific appearance and function. We can switch the type of window and with this we can set the basis for our work. We use the window type "Text editor" to write any texts of comments to our projects or program code in Python language.

Blender has implemented Python compiler with its commands and syntax. [Ble05a] Therefore in many scripts we do not need any Python compiler. We compile and run our Python script by the command "Run Python Script" in menu File of its window.

4. THE CONCEPTION OF PYTHON SCRIPTS IN BLENDER

Each project in Blender is composed of blocks. These blocks are interconnected. Examples of these blocks are Scene (in hierarchy it is at the outside and it contains others objects), World (information about the environment – colors, background, mist, stars, etc.), Object (information about the position, scale and rotation; it contains most of the elements in the scene – meshes, curves, lamps, cameras, etc.), Material (information about the surface of objects; it can contain the texture).

This paper's name is "The creation of simple plants". Each plant is composed of mesh objects. The substance of this script is a mesh creation. The creation of a simple mesh object is described in following steps: creation of a new mesh, setting number of vertices and add them to this mesh, generation a face for these vertices and append to the mesh and the last step is to add the mesh into the Blender scene.

5. THE PYTHON SCRIPT FOR THE CREATION OF SIMPLE PLANTS

Above I have described a simple Python script. Now we will generate a simple plant with some leaves. Each leaf of this plant can be a separated rectangular mesh object.

The leaves does not look realistic (they are rectangular), but it is sufficient for us. This plant is universal – the real faces of leaves give us a material setting. We add material with a texture to all of leaves. This texture will have two layers – the bitmap picture with any leaf and its alpha map. In material menu we set the alpha channel. After this, the final picture with materials of background and world is shown in the Figure 5. Here I use a scanned leaf of a buckhorn.

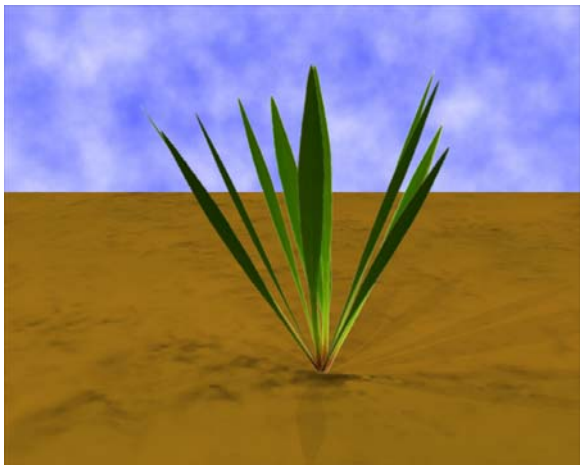


Figure 5. The bitmap – a final render of my scene with a simple plant.

Not all the plants are the same - we want to generate different plants. In our example we can change some parameters – the number and the size of leaves and angles to *x*, *y* and *z* axis. To this we can add a random factor. The randomize part gives us a different face for the same plants.

I have designed a simple window with the use of the buttons. These buttons set parameters for each leaf. The each of leaves is generated in the same place and after this is transformed (scale *x* and *y*, *x* and *z* rotation, random factors) into the final position. We can set these transformations with corresponding buttons. With this we can generate varied plants.

6. CONCLUSIONS

In this paper there, the usage of the Python scripts in Blender has been presented. These subjects are taught at our university within the Computer Graphic.

I have illustrated a programmable simplicity of Python scripts. This example (random plants generator) shows the use of mesh commands in Python language.

7. REFERENCES

[Ble05a] Blender development page at http://www.blender.org

[Cer02a] Černohous, P. Problems with the Python scripts in Blender at http://www.grafika.cz

[Lut03a] Lutz, M., and Ascher, D. Learn Python. Prague, Grada press, 2003.

Ostrava, pp. 267-270, 2004, http://www.ineer.org

[Pok05a] Pokorný, P., and Sysel, M. Using Python script in Blender. In: DAAAM, Opatia, pp. 305-306, 2005, http://www.daaam.com

[Pyt05a] Python official page at http://www.python.org

Virtual Environments for Learners with Special Needs

Selim Balcisoy
Sabanci University
Orhanli - Tuzla
34956, Istanbul, Turkey
balcisoy@sabanciuniv.edu

Elif Ayiter
Sabanci University
Orhanli - Tuzla
34956, Istanbul, Turkey
ayiter@sabanciuniv.edu

ABSTRACT

This paper proposes methods to create educational Virtual Environments for users with no computer and reading skills. The objective is to outline issues that are critical for the success of an educational Virtual Environment: user interface, navigation, interaction and environment design. We propose solutions to improve the usability and visual quality of environments. Two example environments are presented in detail to illustrate our ideas: The education of battered women and a city guide of a historical boulevard in Istanbul.

Keywords

Education, Virtual Environments, User Interfaces.

1. INTRODUCTION

This paper will undertake the task of addressing the issue of providing educational environments for learners with special needs.

The projects presented in this paper are the output of the 'CS450: Computing and Art' course, taught at Sabanci University, an interdisciplinary course involving the collaboration of computer engineering and visual communication design students that are teamed up as one representative of each discipline. These teams are asked to create the content, scenario and design of an immersive Virtual Environment, involving architecture, modeling, animations, image processing, special effects and sound amongst other elements. These Virtual Environments are then implemented into standalone computer applications by the usage of OpenSceneGraph (OSG) [1]. We encouraged our students to give thought to creations that would give possible indicators as to how Virtual Environments can be used as educational environments, especially where little or no literacy is a factor brought into play by the participitant/viewer of the environment. The two examples presented below demonstrate two different approaches to educating in a Virtual Environment without the presence of type. Example # 01 solves the communication problem by voiceover techniques that guide the user to the specific lessons, tasks and components of the architecture, whilst Example # 02 solves the same problem of guidance and communication, i.e. teaching, with the aid of an avatar, i.e. a pedagogic agent.

Posters proceedings ISBN 80-86943-04-6
WSCG'2006, January 30-February 3, 2006
Plzen, Czech Republic.

2. USER INTERFACE AND INTERACTION DESIGN

Our premise is that the users have minimum or no computer knowledge compounded by the overriding factor that they cannot read or write. In their state of stress and nervousness they might not be open to learn any complicated tasks. Furthermore in some cases there might be no trained instructor available on site. Therefore we limited the navigation of the interaction to the automatic selection of an object to its proximity to the mouse. No keyboard is required to navigate and perform selections in the presented environments. The complete environment is accessible just with the movements of the mouse.

Given that it is fairly easy to get lost in Virtual Environments, even for trained users, we have made a special effort to design the navigational components: To address navigation issues we have implemented:

1-) *Limited movement space* around the environment.
2-) *Automated camera placement.*
3-) *Voice/narrative*.

3. CASE STUDIES

Example #1: Educational Virtual Environment for Battered Women

The first environment is on the education of battered women of all education levels, literate as well as illiterate, as a focus area and created a Virtual Environment to educate them about actions they can take to fight against abuse in the family. We have identified the global [2] and local [3] issues on battered women at the beginning of this project. There are a couple of important issues to consider when designing Virtual Environments for not only computer illiterate, but illiterate in the full meaning of the word, target groups.

The interface helps empower women through visual language, provide a safe space for abused women and girls where they can enter and feel secure. The objective is to make women aware of their own power and be in control over their own bodies and lives, that they can in fact make choices in these matters. By showing individual cases, we aim to establish identificatory examples that the user can empathize with as well as make correlations to her own situation, releasing her from the anxiety of feeling confronted with a situation that she is facing alone. (Figure 1)

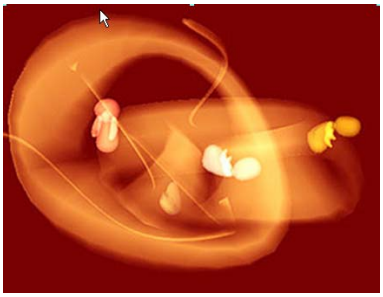


Figure 1: Snap-shots from Virtual Environment to educate battered women. Each fetus contains a different story.

Example #2: The Istiklal Avenue Guide

We created a Virtual Environment that will educate the user in the architecture, history, as well as places of interest on one of the main historical avenues of Istanbul, Istiklal Avenue. The avenue is a pedestrian area that nonetheless has a tramcar running its length. This tramcar has become the symbol of the location and thus we decided to use it as an avatar/pedagogical agent. To this end we employed some minor antropomorphisation to the front of the car, simulating facial features with windshields; bumper and so on and most importantly gave the tramcar a human voice. (Figure 2)

Figure 2: Snapshots from Istiklal Avenue Guide Virtual Environment. The tram is an animated pedagogical agent presenting the history and daily life of the area.

4. CONCLUSION AND FUTURE WORK

In this paper we have proposed a methodology to create interactive 3D Virtual Environments to educate users with no computer and reading skills. Presented example environments provide solutions on interface, interaction and environment design for educational VE's with and without pedagogical agents.

The main result of our experiments is that once carefully designed 3D Virtual Environments can become a promising platform to convey information to people from diverse background and skill

In near future we plan to experiment with a large set of participants from Turkish women rights NGO's and Beyoglu municipality where the second VE takes place to open both environments to public use in selected sites.

5. REFERENCES

[1] OpenSceneGraph www.openscenegraph.org
[2] Amnesty International: http://www.amnesty.org.uk/news/press/14030.shtml, January 10, 2005
[3] "Violence against Women : Terror at Home". Istanbul: Mor Çatı Publications, 1996

Programming Graphical Objects and Information in Engineering Drawings

Algirdas Sokas
Dept of Engineering Graphics,
Vilnius Gediminas Technical
University, Sauletekio al. 11,
LT-10223, Vilnius, Lithuania

algirdas.sokas@fm.vtu.ltT

ABSTRACT

This article introduces you to graphics programming in the AutoCAD environment. Using Visual Basic Application programming language prepared methods and procedures for solving these design problems: attaching, reading and fulfilling mathematical operations with information extended data of a graphical objects. All information about drawing in the AutoCAD system is in the drawing database. We will study it in Drawing Interchange Format (DXF), which is in many graphical systems. Algorithms are formed to attach database record information to a graphical object, to read the information from graphical objects and to create objects specification in the drawing. The prepared example shows framework drawing with specification and database with steel information written in Visual Basic Application programming language. The general purpose of this paper is to contribute to such a debate about possibility of Visual Basic Application graphics programming in engineering education..

Keywords

Data exchange technology, extended data, graphics programming, Visual Basic Application.

1. INTRODUCTION

First computer-aided design (CAD) product Line Drawing System was created by David Evans and Ivan Sutherland in 1969. The first articles about engineering drawings are written about specifications, materials and amounts.

Formulation of problem. Prepare methods, algorithms and procedures for solve these design problems: attach information to a graphical object being drawn, read information from certain graphical objects of a drawing, formation of specification in engineering drawing.

We will study all information about drawing in DXF format, which is in many graphical systems. Data that describe entity is a list. It is made of different DXF group codes. Each such group separated by brackets also form a list from code, dot and meaning. Code defines property, dot is a distinctive sign, and meaning is property's parameter. For example, a list (0 . "LWPOLYLINE") informs that code equal to zero and defines entity type, meaning is entity name. Code "-3" means that the next long list is user extended data. We can append additional data, which are named extended data (xdata), to the graphical entities.

Much has been written in the last years on methods for adaptation of graphic exchange formats: very good adaptation of DXF exchange format structure was presented in geometric modeling of box girder deck for integrated bridge graphical system [Sam03a]; feature-based design of extrusion shapes [Kum04a]; reconstruction of 3D models from 2D orthographic views [Cic04a]; finite element analysis and data creating system with CAD data [Yam05a].

Posters proceedings ISBN 80-86943-04-6
WSCG'2006, January 30-February 3, 2006
Plzen, Czech Republic.

2. OBJECTS AND DATABASE

Programming graphical objects can produce many different graphical results depending from parameters. We can prepare these parameters in another application that is different from the one we draw in. We can draw in the CAD system but access parameters in the database. To prepare VBA project, first we have to create an instance of the application

and to declare a variable that will represent the connection with other application. Second, we have to create a set of statements with declared variables and open database table. Programming with Visual Basic for Application language in the AutoCAD environment. Prepare database with geometrical and other information about concrete graphical object. All concrete information of steel pivot as class, diameter, tension, press all graphical objects we found in the one field of database.

We draw with information from the database. The first procedure selects one graphical object record with all fields in the database of graphical objects' parameters and creates extended data of the new graphical object. There are three types of extended data: a class of steel, length and diameter of a pivot, which are found by a programming method. Information is registered by name "Steel framework". Names and values are attached with codes "1000" and "1040". The next procedure selects graphical objects with extended data in the drawing and writes information to a matrix. The next design problem is formation of specification in an engineering drawing [Sok04a]. We form the matrix with graphical object's information as the steel class, stick's length and diameter. Then we have to sort this information by class, by diameter in the same class, and by length in the same diameter. The next algorithm finds graphical objects with the same length, matching class and diameter in the matrix and calculates their number. This new information is written to a specification table in which the first column has the number of graphical objects in the drawing, the second column has the steel class, stick's diameter and length, and the last column has the corresponding number of these matching objects.

3. RESULTS AND CONCLUSIONS

Prepare methods, algorithms and procedures for solving these design problems: attach information from database to a graphical object being drawn, reading information from certain graphical objects of a drawing and fulfilling sorting and mathematical operations with read information for formation of specification in engineering drawing. Form an algorithm to attach database record information to graphical object: select one graphical object record in the database, create extended data new graphical object, draw programming graphical object depending on extended data and scale in the drawing. Form algorithm to read the information from graphical objects and to collect extended data in the matrix: select graphical objects in the drawing, take first object from beginning, verify whether object has extended data, collect sorting extended data in the matrix.

Prepare example steel framework drawing with specification and database with steel information with program written in VBA language (Figure 1).

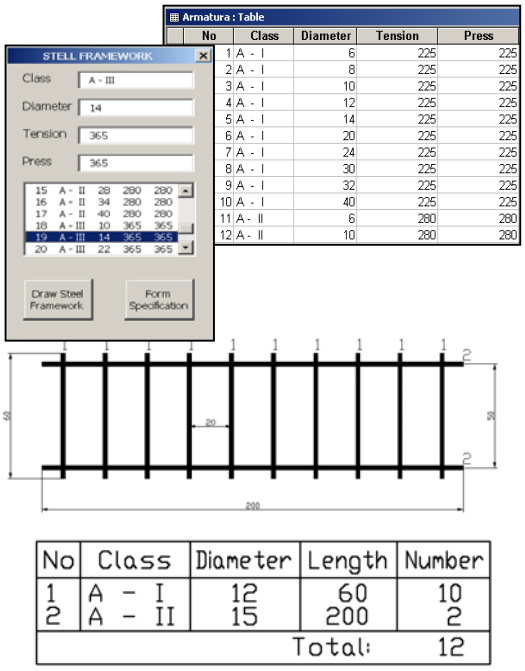


Figure 1. Program menu, database table, drawing and specification.

4. REFERENCES

[Cic04a] Cicek A., Gulesin M. Reconstruction of 3D models from 2D orthographic views using solid extrusion and revolution. Journal of materials processing technology, Vol. 152, No, pp. 291-298, 2004.

[Yam05a] Yamaguchi T., Kawase Y., Nishimura T., Naito H. 2D finite element analysis data creating system with CAD data. Journal of materials processing technology, Vol. 161, No 1-2, pp. 311-314, 2005.

[Kum04a] Kumar S., Prasad S. K. Feature-based design of extrusion process using upper-bound and finite element techniques for extrudable shapes. Journal of materials processing technology, Vol. 155-156, pp. 1365-1372, 2004.

[Sam03a] Sampaio A.Z. Geometric modeling of box girder desk for integrated bridge graphical system. Automation in Construction, Vol. 12, No 1, pp.55-66, 2003.

[Sok04a] Sokas A. Automatized formation of specifications in engineering drawings. The Journal of Polish Society for Geometry and Engineering graphics, Vol. 14, pp. 58-63, 2004.

A Multimodal User Interface Component for an Augmented Reality Mobile User Guidance System

Marco Thum
University of Koblenz-Landau
Working Group Computer Graphics
Campus Koblenz, Germany
mthum@uni-koblenz.de

Thanos Demiris
Content Delivery Systems
Intracom S.A.
Peania, Athens, Greece
dema@intracom.gr

Stefan Müller
University of Koblenz-Landau
Working Group Computer Graphics
Campus Koblenz, Germany
stefanm@uni-koblenz.de

ABSTRACT

In general, user interfaces should be intuitive, self-explanatory and adaptive to various user skills. Especially in augmented reality systems with complex interaction possibilities simple concepts are necessary to guide the user. Therefore, we extend the ideas of the traditional WIMP metaphor (Windows, Icons, Menus, Pointer) with the capabilities of multimodal interfaces where multiple ”human” communication channels are used as input data for navigation, orientation and interaction.
In this work we will present the adaption of the user interface of an existing mobile augmented reality system for cultural heritage to multimodal interaction. We present the concepts that led us to the decision for the use of speech and capturing of hand movements by means of an inertial tracker, as well as the implementation aspects of the initial prototype. User evaluation trials will prove our approach.

Keywords: Augmented reality, mixed reality, graphical user interfaces, multimodal user interfaces, human computer interaction, software ergonomics, cultural heritage.

1 INTRODUCTION

The system is embedded as a client in the architecture of the intCulture [2] solution for the management and dissemination of multimedia content related to cultural heritage, featuring an extensive back-office and a set of supported client platforms. One of them is the augmented reality mobile guide, for which we implemented the multimodal interaction system presented herein. It integrates the ARBrowser developed at Fraunhofer IGD and featuring markerless tracking [8]. Because of less training necessity we decided for speech as the primary input device incorporating a speaker-dependent recognition engine. A sub-notebook, a video-see-through head mounted display resp. binoculars, an one-ear headset, a GPS sensor and an electronic compass are forming the hardware part of the system. An ordinary desktop mouse resp. a wireless remote control with inertial tracker have been chosen as secondary input devices.

2 RELATED WORK

We examined the usage of interaction metaphors and input devices of several kinds of AR user interfaces addressing various application domains, tasks and varying user skills. Guidelines and hypotheses have been derived from [1, 4, 9] et al.

Posters proceedings, ISBN 80–86943–04–6
WSCG'2006, January 30 – February 3, 2006
Plzen, Czech Republic.

3 METHODOLOGY

We continued defining hypotheses proving the general user dialogue and guidance principles laid down by [3]. Characteristics of multimodal interfaces [7] were also evaluated in the user trials. Several use cases followed which fit possible user interactions and needs observed by [5].

3.1 Application Scenario

Visible elements can be dynamically arranged all over the screen. A thumbnail view presents strictly location-sensitive guiding content like contained, adjacent or related site elements. We integrated object related information access in augmented menus aligned as translucent overlays on top of the augmented image as fig. 1 demonstrates. The inertial tracked input device was additionally used for recording and analysis of hand movements to deliver appropriate feedback to the user's actions. The speech dictionary is extended dynamically with keywords of recently visited exhibits.

3.2 System Architecture

For the detailed system design we extracted basic functionality needs out of the scenario and sorted them into patterns and components in accordance to [6]. Almost every component of the architecture is entirely configured via XML. Each input device has to be initialized using a separate configuration file and setting event filters. Events are processed as shown in fig. 2. System messages are logged via a centralized

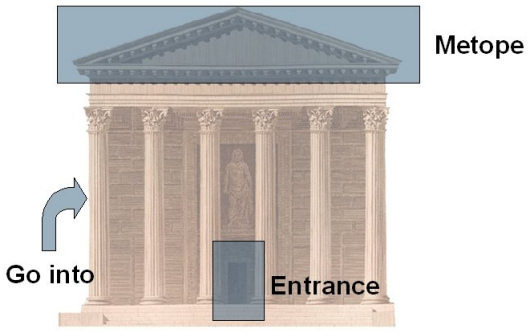


Figure 1: Translucent rectangles indicate additional information on exhibit parts. Handlers like pointers lead to manipulations on the exhibit (concept).

global mediator system and appropriate feedback is presented as audio or text.

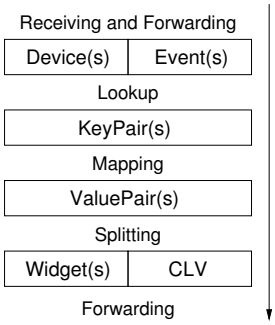


Figure 2: Events are dispatched to widgets by comparing tuples to a key map. Widgets interpret events as Common Layer Values (CLV) which are independent from the event description sent by the device. This leads to a highly configurable and extensible event management.

3.3 User evaluation

For the initial trials we separated 14 users into 2 groups. The first group was requested to use a head-mounted display and an ordinary desktop mouse. The second one interacted with a wireless, inertial tracked remote control and used binoculars for the video output. The speech recognition engine was trained to a greek male and greek female voice. A german male profile has been set up for almost acentless speakers. The test system has been arranged locally. A questionnaire should prove the hypotheses deployed in the design. A short extract of the evaluation results is given in fig. 3.

4 CONCLUSION & FUTURE WORK

On the basis of the outcome of the evaluation a set of adaptations will take place, which will lead to the second prototype of the system. Randomly selected visitors of a selected site (either the archaeological

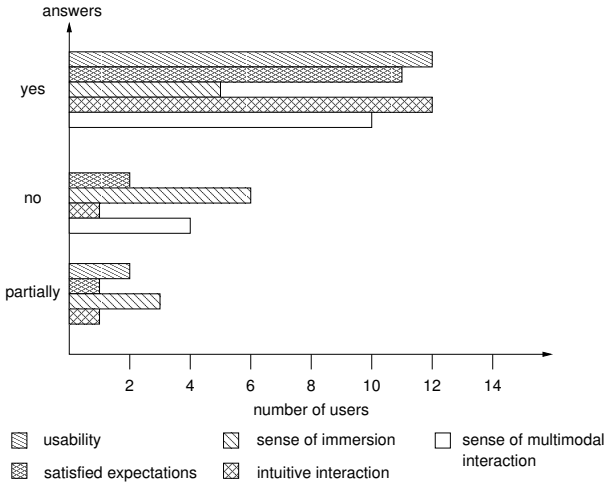


Figure 3: Evaluation results of guidelines and hypotheses.

site of ancient Pompeii in southern Italy or ancient Olympia in Greece) will evaluate the improved system.

5 ACKNOWLEDGEMENTS

We would like to thank Nikos Pronios and Nikos Ioannidis as representants of Intracom S.A., Peania, Greece, for the support and funding of this project.

REFERENCES

[1] O. Achenbach, D.-I. S. Göbel, and O. Schneider. GEIST – A Narrative AR Learning Application. *INIGRAPHICS – computer graphik topics*, 16(3), 2004.

[2] T. Demiris, V. Vlahakis, and N. Ioannidis. intCulture: Location-based Multiplatform Publishing of Cultural Heritage Information. In *Proceedings of the ICHIM 2004*, pages CD–ROM paper C2, 2004.

[3] European Committee for Standardization. Ergonomic requirements for office work with visual display terminals (EN ISO 9241-(10 &13)). Technical report.

[4] T. Höllerer, S. Feiner, T. Terauchi, G. Rashid, and D. Hallaway. Exploring MARS: Developing Indoor and Outdoor User Interfaces to a Mobile Augmented Reality System. *Computer & Graphics*, 23(6):779–785, 1999.

[5] S. Hsi. The Electronic Guidebook: A Study of User Experiences using Mobile Web Content in a Museum Setting . In *Proceedings of the IEEE International Workshop on Wireless and Mobile Technologies in Education* , 2002.

[6] A. MacWilliams, T. Reicher, G. Klinker, and B. Bruegge. Design Patterns for Augmented Reality Systems. In *IUI-CADUI Workshop on Exploring the Design and Engineering of Mixed Reality Systems*, volume 91, 2004.

[7] S. Oviatt. Multimodal Interfaces. In Julie A. Jacko and Andrew Sears, editor, *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, chapter 14, pages 286–304. 2002.

[8] D. Stricker. Tracking with Reference Images: A Real-Time and Markerless Tracking Solution for Out-Door Augmented Reality Applications. In *Proceedings of Conference on Virtual Reality, Archeology, and Cultural Heritage*, pages 77–82, Darmstadt, Germany, 2001.

[9] J. Yang, W. Yang, M. Denecke, and A. Waibel. Smart Sight: A Tourist Assistant System. In *ISWC*, pages 73–78, 1999.

ICP fitness analysis for 3D scan data matching

J.S.M. Vergeest, Y. Song
Delft University of Technology
Landbergstraat 15
NL-2628 CE Delft, The Netherlands
y.song@tudelft.nl

ABSTRACT

Full automation of the registration of 3D scan data is, in general, still an unsolved problem. If supplementary data is provided, either by human assistance, or by additional devices, the registration process can be completed. We have analyzed under which conditions supplementary data is required, where the conditions are specified as subsets of configuration space. As a matching tool an ICP-based (iterative closest point) algorithm was deployed. A point in configuration space represents the amount of translation and rotation needed to obtain a match between 3D scans of two partial overlapping surfaces. As a function of the coordinates in configuration space, we determine the successfulness of the algorithm and hence whether additional information is required. Our shape matching method does not rely on pre-computation of surface invariants, nor on the identification of shape features. The possible application of the analysis results for practical 3D scanning purposes are described.

Keywords
Freeform shape design, 3D scanning, registration, automation

1. INTRODUCTION

The reconstruction of a 3D geometric model of a physical model from measurements of points on its surface is a widely applied process. Although theoretically a fully automatic registration process can be exclusively based on the individual views it can still be opted to include supplementary data about the initial relative or absolute poses of the views for the following reasons: 1) The number of wrong pair-wise matches can be reduced, making the whole process more robust, 2) the algorithm can be faster as significant portions of the transform spaces need not to be considered in a search. In this paper we explore when information would be required in order to obtain the initial transform. In section 2 we provide the problem statement and the numerical setup. In section 3 we present a few numerical results and some examples.

Posters proceedings ISBN 80-86943-04-6
WSCG'2006, January 30-February 3, 2006
Plzen, Czech Republic.

2. PROBLEM DESCRIPTION AND NUMERICAL PROCEDURE

We consider the local registration of two (out of possibly many) datasets, *A* and *B*, each representing a portion of the surface of an object.

We define a matching criterion as a discrete mapping $M(X, Y) \rightarrow \{0,1\}$, where X and Y are geometric sets. $M(X, Y) = 0$ means that X and Y match, and $M(X, Y) = 1$ means that X and Y do not match. In the literature there exist several real-valued distance measures of geometric sets [Hub2003] and threshold versions of them.

For this numerical analysis we assume to have pre-knowledge about the optimal relative pose of *A* and *B*, which we set to the identity pose *I* without losing generality. The input to the problem are the sets *A* and *T*(*B*), where $T \in 3^3 \times \mathrm{SO}(3)$ is a known transform, consisting of a rotation specified by a 3×3 matrix of the orthogonal group SO(3) followed by a translation specified by a vector $(d_x, d_y, d_z)^T$ in 3^3.

$$T=\begin{pmatrix} \cos\alpha\cos\beta & \cos\alpha\sin\beta\sin\gamma-\sin\alpha\cos\gamma & \cos\alpha\sin\beta\cos\gamma+\sin\alpha\sin\gamma & d_x \\ \sin\alpha\cos\beta & \sin\alpha\cos\beta\sin\gamma-\cos\alpha\cos\gamma & \sin\alpha\sin\beta\cos\gamma-\cos\alpha\sin\gamma & d_y \\ -\sin\beta & \cos\beta\sin\gamma & \cos\beta\cos\gamma & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where the angles α, β and γ can be interpreted following the *x-y-z* fixed-angles convention [Cra1989]. Wanted is the transform *S* such that $M(A, S(T(B))) = 0$.

The search method that we use is based on Levenberg-Marquardt (LM) minimization of an ICP distance function of *A* and *B* [Lev1944, Lou2003].

3. CONFIGURATION SPACE ANALYSIS

For the numerical test we used a test set *A* of unordered points obtained by scanning of a physical object. Set *A* consists of 2307 points and represents a surface portion of about 30×20 mm, see Fig. 1. The spacing between the points was typically 0.5mm.

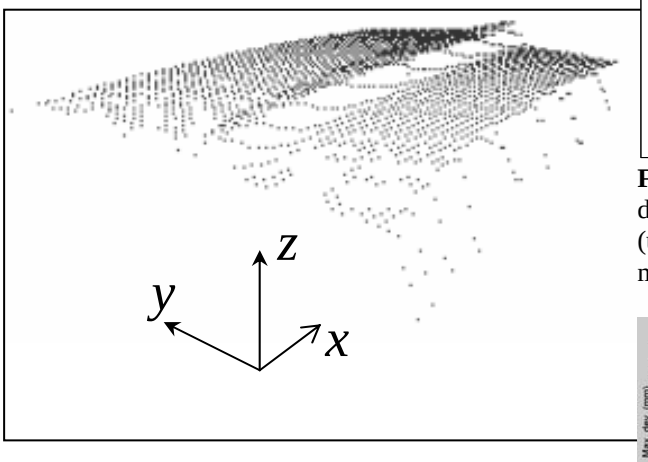


Figure 1. Unordered set *A* of points

Set *B* was taken identical to *A*, implying that the algorithm might achieve a perfect match of *A* and *T*(*B*). We also disregard the problem of finding the overlap region and of possible differences between *A* and *B* 's scanning accuracy and resolution. A general impression of the performance of the algorithm is shown in Figure 2. In Figures 3 and 4 the resulting deviation is shown as a function of two variables, keeping the remaining 4 variables fixed to zero. The results indicate that a combination of rough pre-alignment and sampling can support the registration process.

REFERENCES

[Cra1989] J.J. Craig (1989), "Introduction to robotics: mechanics and control". Addison-Wesley,.

[Dyn2005] Dynash Research Project (2005), Delft University of Technology, http://www.dynash.org.

[Hub2003] D.F. Huber and M. Hebert (2003), "Fully automatic registration of multiple 3D data sets". Image and Vision Computing 21, pp 637-650.

[Lev1944] K. Levenberg (1944), "A method for the solution of certain non-linear problems in Least Squares. Quarterly of Applied Mathematics, 2(2):164-168.

[Lou2003] M.I.A. Lourakis, A.A. Argyros (2003), "Efficient 3D camera matchmoving using marker-less segmentation-free plane tracking. technical Report 324, Institute of Computer Science, FORTH, Heraklion, ftp://ftp.ics.forth.gr/tech-report

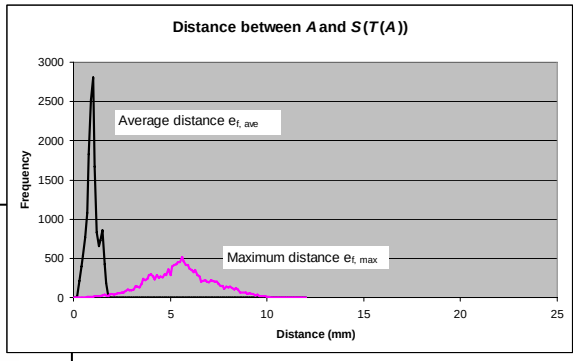


Figure 2. Distribution of maximum and mean distance (15625 entries) between point clouds before (upper picture) and after (lower picture) the ICP matching process.

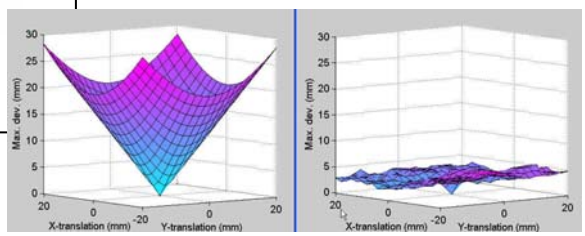


Figure 3. Distance between the point clouds before and after matching. Max deviation between *T*(*A*) and *A* (left) and between *S*(*T*(*A*)) and *A* (right) for initial translations in *x*- and *y*-directions. The remaining variables are fixed to zero.

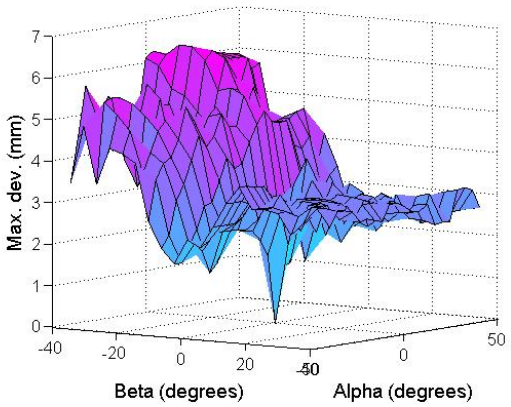


Figure 4. Max. deviation after matching as function of large initial α- and β-rotation, with remaining variables fixed to zero.

A Probabilistic Approach for Object Recognition in a Real 3-D Office Environment

Michael Wünstel
Universität Bremen
Fachbereich Mathematik und Informatik
Postfach 330 440
28334 Bremen, Germany
wuenstel@informatik.uni-bremen.de

Thomas Röfer
Universität Bremen
Fachbereich Mathematik und Informatik
Postfach 330 440
28334 Bremen, Germany
roefer@informatik.uni-bremen.de

ABSTRACT

The scenario used focuses on object recognition in an office environment scene with the goal of classifying office equipment that is located on a table. The recognition system operates on three-dimensional point-clouds of objects on a loosely covered table where no previous information about the precise position of the table is given. As the point-clouds do not cover the complete objects and the data is noisy, especially for smaller objects a robust detection of special features is difficult.

The workflow employed is a three step process: In a first step the table plane is detected and the point clouds of the objects are extracted from the surface. In the second step an object-oriented bounding-box is calculated to get the geometric dimensions, i.e. the properties measured. During a learning phase these simple features are used to calculate the parameters of Bayesian networks. The trained networks are used in the third step, i.e. the classification step. The dimensions of an unknown object form the input for a Bayesian network that yields the most probable object type.

Keywords

Object Recognition, Cognitive Vision, Bayesian Network, Laser Range Data.

1. INTRODUCTION

There are many strategies for object recognition tasks that can be distinguished, e.g., concerning their perception process, namely object recognition strategies operating on 2-D density images or on 3-D range images. Range data has been used for object recognition for over two decades [1]. In this paper an approach is presented, in which the identification is done using just the dimensions of the objects together with general scene knowledge. This means on the one hand that certain objects correlate with certain dimensions, and on the other hand their presence correlates with the given scene. The recognition process is based on these simple but robust features as the extraction of three-dimensional special features is difficult due to occlusions, data noise, and the lack of measuring points especially for small objects.

Posters proceedings ISBN 80-86943-04-6
WSCG'2006, January 30-February 3, 2006
Plzen, Czech Republic.

2. System

A laser range finder that is mounted on a pan-tilt unit scans the scene. During a vertical scan of the pan-tilt unit several layers of scan points are collected and afterwards combined to form a three dimensional depth image. Afterwards the table plane is detected and the objects lying on it are segmented. In the next step an object oriented bounding-box is calculated that yields the dimension measures.

Sensor

As the laser sensor is mounted on a pan-tilt unit, it provides depth information from a viewpoint that is about 1.35 m above the ground (see Figure 1). The system has a horizontal resolution of 0.5° and the scanner is tilted in 0.5° respectively 1° steps.

Figure 1: Table scene with scanning equipment

Segmentation of Objects

As there is no specific information about the table, the exact position of the table top has to be determined separately. This information gives the vertical base level for the object segmentation step. For the plane detection the normal of every point is calculated [2] and connected regions are determined. The object segments are then calculated by a 3-D point density algorithm. This can be done from different positions (see Figure 2).

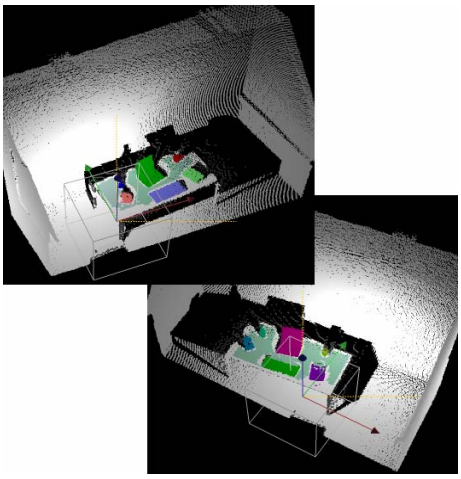


Figure 2: Detected table top and segmented objects from two different positions

Geometric Dimensions

The bounding-box that gives the measurement of an object is calculated iteratively using an optimization function that describes the quality of the actual bounding-box depending on two terms. The first (point-based) term describes the sum of the distances of the points to the bounding-box. The second (side-based) term describes the fitness for a subset of points to each separate side. Beginning with the axes-oriented bounding-box this function is minimized using the downhill simplex method (see Figure 3).

3. Recognition

The key method for the recognition process is the use of Bayesian networks:

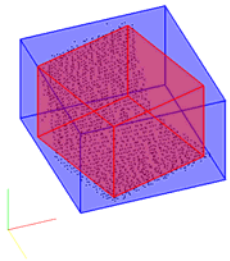


Figure 3: Initial axes-oriented bounding-box (red) and resulting object-oriented bounding-box (blue) for a notebook

Learning Phase

The dimensions of an object category are described by naive continuous Bayesian networks. Every object class corresponds to a single Bayesian Network that is parameterized by its mean and variance values. These values are determined analytically using the values from a training set.

Classification

Given the dimensions of an object, the grade of membership to each class is determined. Therefore the probability density function of each object class, resulting from the training phase, is integrated over an interval around the given evidence. The object is classified to the class with the highest probability value.

4. Results

The eight object classes used for the experiments are the following ones: book, bottle, coffeepot, keyboard, mug, notebook, phone, and hole-punch. The training set consists of ten different well-segmented shots of theses objects, the test set of five. For half of the objects the classification was always correct, for the other half up to two misclassifications occurred.

5. ACKNOWLEDGMENTS

This project is supported by the Deutsche Forschungsgemeinschaft, DFG, through the inter-disciplinary Transregional Collaborative Research Center “Spatial Cognition: Reasoning, Action, Interaction”.

Parts of this work go back to the work of Andreas Meyer. Other parts were developed with support of Christian Kanthak. Thanks for their effort.

6. REFERENCES

[1] Paul J. Besl and Ramesh C. Jain. Three-dimensional object recognition. ACM Computing Surveys, 17(1):75–145, March 1985.

[2] I. Stamos and P. Allen. 3-D Model Construction using Range and Image Data. CVPR-00, pages 531–536, 2000. IEEE.

Automatic Control Point Segmentation and Localization for Online Camera Calibration

Chunrong Yuan
Fraunhofer FIT
Schloss Birlinghoven
53754 Sankt Augustin, Germany
yuan@computer.org

ABSTRACT

Camera calibration is an essential step for 3D object recognition and reconstruction. In this work, we propose a novel method for the automatic localization of control points and present a simple and yet robust online camera calibration system. First a planar calibration pattern with circular control points is designed. These points are then automatically segmented and localized by integrating information resulted from two complementary segmentation methods: region extraction and contour detection. After the calibration pattern has been shown to the system at a few locations, both the intrinsic and extrinsic camera parameters can be determined. The main advantage of this approach is that the metric measurement of the calibration pattern in the image plane is done purely automatically. We have implemented the whole system on a windows platform. Tests with USB cameras in different configurations and applications show the efficiency as well as the accuracy of the proposed approach.

Keywords

Camera calibration, image segmentation, contour detection, automatic point localization, parameter estimation

1. INTRODUCTION

Camera calibration is the process of determining the set of unknown parameters of a certain camera. It is generally done by using a suitable calibration pattern. The set of parameters can be computed by inferring the relationship between the 3D world coordinate of some reference points on the calibration pattern and their 2D image coordinate. The whole calibration procedure includes the definition of a camera model, the design of a calibration pattern with a definite geometric configuration of control points, the acquisition of a few images of the calibration pattern to obtain the locations of the control points in the images, the determination of the camera parameters by fitting the extracted points to the camera model and solving a system of equations.

Onrsdqr oqnbddc‰fr ‰RAM 7/,75832,/3,5
WSCG'2006, January 30-February 3, 2006
Plzen, Czech Republic.

There are numerous methods for camera calibration presented in the literature [Tsai87a, Hei97a]. While many of them focus on modeling the image formation process and solving the camera projection matrix, we discuss the whole pipeline of camera calibration and pay particular attention to the measurement procedure of the 2D image coordinate of the reference points. Unlike [Zha00a], where squares are used as control points, we use circles for the purpose of more robust localization of the control points under unfavorable lighting conditions and heterogeneous backgrounds.

2. AUTOMATIC LOCALIZATION OF CONTROL POINTS

A calibration pattern with some circular points on it that act as control points has been designed, as is shown in Fig. 1. Altogether there are 70 circular points whose diameter is equal to 23 mm. An input image **f** of the calibration pattern is first segmented separately using two different approaches, namely threshold based region extraction and contour detection. Then an algorithm has been developed to integrate the information resulted from the two segmentation approaches. Illustrated in Fig. 2 is the output image with the extracted control points shown

as white points in the centre of each circular contour. As can be seen, all the circular points have been segmented properly despite the fact that the background of image $\mathbf{f}$ is heterogeneous.

3. CAMERA MODEL & ESTIMATION

Using a pinhole camera model, A 3D point $\mathbf{P} = [x_w, y_w, z_w, 1]^T$ is mapped to its 2D image point $\mathbf{p} = [x_i, y_i, 1]^T$ with $\lambda \mathbf{p} = \mathbf{MP}$, where $\mathbf{M} = \mathbf{K}[\mathbf{R} \quad \mathbf{t}]$, and $\mathbf{K} = \begin{bmatrix} f_x & 0 & cc_x \\ 0 & f_y & cc_y \\ 0 & 0 & 1 \end{bmatrix}$.

Matrix $\mathbf{M}$ has 10 unknowns, 4 intrinsic parameters f_x, f_y, cc_x, cc_y and 6 extrinsic ones (three embedded in $\mathbf{R}$ and three in $\mathbf{t}$). To model lens distortions, we use five parameters (three radial and two tangential distortion coefficients) $k_i, i = 1,...5.$

From the 3D and 2D correspondences, a linear solution can be obtained for the 4 intrinsic camera parameters. Using the Levenberg-Marquardt algorithm, k_i and the 6 extrinsic camera parameters that are associated with each input image of the calibration pattern can be estimated.

4. EXPERIMENTAL RESULTS

The proposed approach has been implemented on a PC operated under windows XP. The algorithm is tested with a pair of Logitech USB cameras, which are placed parallel on a stereo rig. Calibration of the two cameras (one left camera and one right camera) can be done simultaneously within a few minutes.

After obtaining the intrinsic and extrinsic camera parameters, we project the 3D world coordinate of the control points back to the image plane. The deviation is within 0.25 pixels in the image frame.

With the extrinsic camera parameters of both cameras estimated, the relative pose of the right camera in the left camera coordinate system can be computed. It has been found out that the computed relative 3D pose is quite stable and agrees with the actual configuration of the two cameras.

The calibration algorithm has also been integrated into an object tracking system. The goal is to calculate the 3D pose of some marker objects observed by the calibrated cameras. A satisfying tracking accuracy has been achieved (the deviation of the reconstructed markers in 3D is within 1 cm by translation and 3 degrees by rotation).

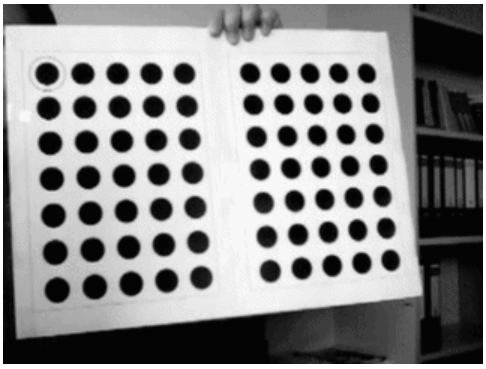


Figure 1. Input image of the calibration pattern.

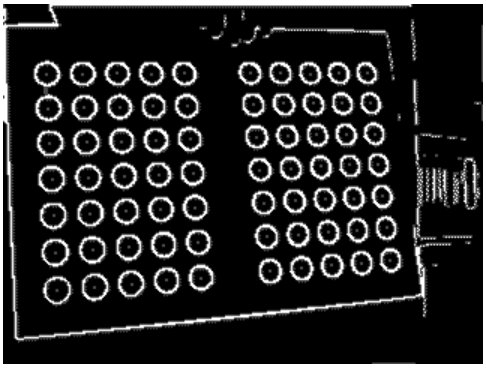


Figure 2. Results with control points localized.

5. CONCLUSIONS

We have presented a new and easy online camera calibration technique based on a planar calibration pattern with circular control points. The proposed approach concentrates mainly on the automatic extraction of 2D control points. By analyzing the calibration images in an unsupervised way, a precise measurement of the reference points can be achieved. The calibration system has been implemented on a windows platform and tested with USB cameras. The efficiency and accuracy of the proposed methodology have been demonstrated by examples.

6. REFERENCES

[Hei97a] Heikkila, J., and Silven, O. A four-step camera calibration procedure with implicit image correction: in IEEE Computer Society Conference on Computer Vision and Pattern Recognition (ICPR'97), pp. 1106-1112, 1997.

[Tsai87a] Tsai, R. Y. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses: IEEE Journal of Robotics and Automation, vol.3, no. 4, pp323-344, 1987.

[Zha00a] Zhang, Z. A flexible new technique for camera calibration: IEEE Trans. PAMI, vol. 22, no. 11, pp 1330-1334, 2000.