



SIGRAD
2015

Proceedings of SIGRAD 2015

June 1ST and 2ND
Stockholm, Sweden

Edited by

Christopher E. Peters and Lars Kjelldahl

The publishers will keep this document online on the Internet - or its possible replacement - from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for your own use and to use it unchanged for any noncommercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law, the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to <http://www.ep.liu.se/>.

Linköping Electronic Conference Proceedings No. 120
Linköping University Electronic Press
Linköping, Sweden, 2015
ISBN: 978-91-7685-855-4
ISSN: 1650-3686 (print)
ISSN: 1650-3740 (online)
http://www.ep.liu.se/ecp_home/index.en.aspx?issue=120

FOREWORD

Welcome to SIGRAD 2015 –

the annual meeting of the Swedish Computer Graphics Association (SIGRAD) – taking place in 2015 at KTH, the Royal Institute of Technology in Stockholm.

The association's mission is to be a meeting point for researchers and industry professionals who are interested in computer graphics and adjacent areas, such as visualization and human-computer interaction (HCI). In recent years, the association's activities were focused on the organization of the annual conference, which attracts a growing number of participants. The rapid development of computer graphics technologies and the acknowledged need for visual computing solutions has led to a growing interest in graphical computing in both commercial and academic areas. Started in Sweden in 1976, SIGRAD has become an annual appointment for the Nordic community of graphics and visual computing experts with a broad range of backgrounds. Through more than three decades, SIGRAD has offered a forum to present and disseminate new technological results, new paradigms and new visions advancing the state-of-the-art of visual computing.

SIGRAD 2015 offers a scientific program representing a cross-section of research in a multitude of domains related to visualization and human computer interaction. In addition to the regular submissions concerning general computer graphics practices, this year's conference also investigates synergies between industry, pedagogical practitioners and academic researchers through a series of presentations, keynote speeches and special sessions. The selection of 18 papers presented at the conference come from researchers in 6 different countries. These papers range from those concerning general computer graphics practices to a subset examining innovative pedagogical practices that may benefit from the use of visualizations and game technologies. The extended participation of students at all levels of academia in research has been encouraged this year through the organisation of a special session involving 7 papers that have been first-authored by students studying at Master's Degree level. Peer reviewing was conducted by a highly qualified Program Committee consisting of 24 reviewers from 7 different countries. Each paper was reviewed, on average, by three reviewers from the committee, with the majority of papers receiving four reviews.

We especially welcome our invited speakers: Tino Weinkauf (Professor of Visualization at KTH) giving a keynote entitled "Flow maps - Benefits, Problems, Future Research". John Fuller (Senior Producer at Avalanche Studios) who gives a keynote entitled "In Pursuit Of Reality. How procedural approaches lead to emergent behaviour and increased immersion". And Tommy Palm (CEO of Resolution Games) who gives a keynote entitled "News from the trenches. The frontier of VR games in 2015".

Finally, we wish to thank Björn Thuresson and Henrik Edlund at the Visualisation (VIC) Studio, KTH for their great efforts and generous support in helping to organise and host the event.

The SIGRAD 2015 organisers

VENUE

KTH ROYAL INSTITUTE OF TECHNOLOGY

KTH, established in 1827, is one of Europe's top schools for science and engineering, graduating one-third of Sweden's undergraduate and graduate engineers in the full range of engineering disciplines. Enrolment is about 17,500 students, of which about 1,400 are pursuing PhD studies. KTH is the coordinator of the Swedish e-Science Research Center (SeRC), a national e-Science activity funded by the Swedish Research Council with some 3 Million Euro per year for an initial period of 5 years, KTH leads a consortium consisting of Stockholm University, Karolinska Institute, Linköping University, and KTH.

VISUALIZATION STUDIO (VIC)

The Visualisation (VIC) studio is a knowledge arena for visualisation stakeholders in the Stockholm region, with the main task of linking business, government, and academia. VIC Stockholm currently has more than 40 member companies and organisations, arranging seminars, workshops, lunch meetings, courses and forum days, to develop and strengthen the domain and create new collaborations and business cases. It also offers a strong link to research and teaching, including on-site R&D in the Visualisation studio and R&D applications. Research is focused on the development and use of innovative visualization and interaction technologies pushing the boundaries of scientific, pedagogic and commercial endeavour. The studio provides extensive equipment for visual demonstrations and managing multimodal interactions with users.



PROGRAM

JUNE 1ST

12.00 Registration

12.15 **Keynote:** Flow maps - Benefits, Problems, Future Research
Tino Weinkauf, Prof, Chair of Visualization, KTH

13.00 **Paper session 1**

Analysis of Visual Arts Collections. *Hermann Pfleiderer and Thomas Ertl*

Multiple Material Meshes for Erosion Simulation. *Věra Skorkovská and Ivana Kolingerová*

Exact Bounding Spheres by Iterative Octant Scan. *Thomas Larsson*

Complete Quadtree Based Construction of Bounding Volume Hierarchies for Ray Tracing. *Ulises Olivares, Arturo García and Félix F. Ramos*

14.30 Coffee break

14.50 **Paper session 2**

Visualizing Single-camera Reprojection Errors using Diffusion. *Zlatko Franjicic and Morten Fjeld*

Civic Participation and Empowerment through Visualization. *Samuel Bohman*

Exploring Time Relations in Semantic Graphs. *Ralph Wozelka, Mark Kröll and Vedran Sabol*

Advanced Visualization Techniques for Laparoscopic Liver Surgery. *Dimitrios Felekidis, Peter Steneteg and Timo Ropinski*

15.30 **Education session**

Teaching OpenGL and Computer Graphics with Programmable Shaders. *Johan Nysjö and Anders Hast*

Introducing Computer Game Technologies in a Mathematical Modelling and Simulation Course. *Christopher E. Peters and Johan Hoffman*

A Case Study in Expo-Based Learning Applied to Information Visualization. *Mario Romero*

PROGRAM

JUNE 1ST (continued)

- 16.15 **Keynote:** In Pursuit Of Reality. How procedural approaches lead to emergent behaviour and increased immersion
John Fuller, Senior Producer, Avalanche Studios
- 17.00 Demonstration and posters
- 17.45 Dinner and social event

JUNE 2ND

- 09.00 **Keynote:** News from the trenches. The frontier of VR games in 2015
Tommy Palm, Entrepreneur and Game Designer, CEO at Resolution Games
- 09.45 Coffee break
- 10.00 **Paper session 3**
- The Effects of Peripheral Use on Video Game Play. *Kimberly Stinson and Niclas Bohman*
- Playful Advertising: In-Game Advertising for Virtual Reality Games. *Xiaopeng Li and Mario Romero*
- Triangulation painting. *Max Pihlström, Anders Hast and Anders Brun*
- Real-Time Fluids - Optimizing Grid-Based Methods. *Jaime A. Losada, Eike F. Anderson and Oleg Fryazinov*
- Visualizing the Effects of Public Transportation Growth on Urban Demographics. *Anders Bea, Dan Cariño, Erik Dahlström, Niclas Ericsson, Julia Gerhardsen, Evert Lagerberg, Fiona Stewart and Mario Romero*
- Facial hair and trustworthiness in virtual faces: Towards an evaluation study. *Evmorfia Kalogiannidou and Christopher E. Peters*
- WOW-A-Cluster! A Visual Similarity-Based Approach to Log Exploration. *James E. Twellemeyer, Arjan Kuijper and Jörn Kohlhammer*
- 11.30 Summary and panel discussion
- 12.00 Close

ORGANIZATION

GENERAL CHAIRS

Christopher E. Peters, KTH Royal Institute of Technology
Lars Kjelldahl, KTH Royal Institute of Technology
Björn Thuresson, KTH Royal Institute of Technology
Mario Romero, KTH Royal Institute of Technology
Morten Fjeld, Chalmers University of Technology

PROGRAM CHAIRS

Thomas Larsson, Mälardalen University
Lars Kjelldahl, KTH Royal Institute of Technology

EDUCATION CHAIRS

Michael Doggett, Lund University
Christopher E. Peters, KTH Royal Institute of Technology

INTERNATIONAL PROGRAMME COMMITTEE

Tomas Akenine-Möller, Lund University, Sweden
Eike F. Anderson, Bournemouth University, UK
Nils Andersson, EON Reality, Sweden
Ulf Assarsson, Chalmers University of Technology, Sweden
Jörgen Björklund, Paradox Interactive, Sweden
Alan Chalmers, Warwick University, UK
Matt Cooper, Linköping University, Sweden
John Dingliana, Trinity College Dublin, Ireland
Thomas Ertl, University of Stuttgart, Germany
John Fuller, Avalanche Studios, Sweden
Eduard Gröller, TU Vienna, Austria
Kai-Mikael Jää-Aro, Södertörns Högskola, Sweden
Sophie Jörg, Clemson University, USA
Andreas Kerren, Linnaeus University, Sweden
Fotis Liarokapis, Masaryk University, Czech Republic
Carol O' Sullivan, Disney Research, USA/Trinity College Dublin, Ireland
Tommy Palm, Resolution Games, Sweden
Timo Ropinski, Ulm University, Germany
Stefan Seipel, Uppsala University, Sweden
Yngve Sundblad, KTH Royal Institute of Technology, Sweden
Veronica Sundstedt, Blekinge Institute of Technology, Sweden
Jonas Unger, Linköping University, Sweden
Rüdiger Westermann, TU Munich, Germany
Anders Ynnerman, Linköping University, Sweden

Table of Contents

Paper Session 1

Analysis of Visual Arts Collections <i>Hermann Pfleger and Thomas Ertl</i>	1
Multiple Material Meshes for Erosion Simulation <i>Věra Skorkovská and Ivana Kolingerová</i>	5
Exact Bounding Spheres by Iterative Octant Scan <i>Thomas Larsson</i>	9
Complete Quadtree Based Construction of Bounding Volume Hierarchies for Ray Tracing <i>Ulises Olivares, Arturo García, and Félix F. Ramos</i>	13

Paper Session 2

Visualizing Single-Camera Reprojection Errors Using Diffusion <i>Zlatko Franjic and Morten Fjeld</i>	17
Civic Participation and Empowerment through Visualization <i>Samuel Bohman</i>	20
Exploring Time Relations in Semantic Graphs <i>Ralph Wozelka, Mark Kröll and Vedran Sabol</i>	24
Advanced Visualization Techniques for Laparoscopic Liver Surgery <i>Dimitrios Felekidis, Peter Steneteg and Timo Ropinski</i>	28

Education Session

Teaching OpenGL and Computer Graphics with Programmable Shaders <i>Johan Nysjö and Anders Hast</i>	32
Introducing Computer Game Technologies in a Mathematical Modelling and Simulation Course <i>Christopher E. Peters and Johan Hoffman</i>	35
A Case Study in Expo-Based Learning Applied to Information Visualization <i>Mario Romero</i>	38

Paper Session 3

The Effects of Peripheral Use on Video Game Play <i>Kimberly Stinson and Niclas Bohman</i>	42
Playful Advertising: In-Game Advertising for Virtual Reality Games <i>Xiaopeng Li and Mario Romero</i>	46

Triangulation Painting <i>Max Pihlström, Anders Hast, Anders Brun</i>	49
Real-Time Fluids – Optimizing Grid-Based Methods <i>Jaime A. Losada, Eike F. Anderson and Oleg Fryazinov</i>	53
Visualizing the Effects of Public Transportation Growth on Urban Demographics <i>Anders Bea, Dan Cariño, Erik Dahlström, Niclas Ericsson, Julia Gerhardsen, Evert Lagerberg, Fiona Stewart and Mario Romero</i>	56
Facial Hair and Trustworthiness in Virtual Faces: Towards an Evaluation Study <i>Evmorfia Kalogiannidou and Christopher E. Peters</i>	59
WOW-A-Cluster! A Visual Similarity-Based Approach to Log Exploration <i>James E. Twellemeyer, Arjan Kuijper and Jörn Kohlhammer</i>	61

Analysis of Visual Arts Collections

H.Pflüger & T.Ertl

Institute for Visualization and Interactive Systems (VIS), University of Stuttgart

Abstract

In this paper, we introduce a projection technique that aims to place points representing individual images in a two-dimensional visualization space so that proximity in this space reflects some sort of similarity between the images. This visualization technique enables users to employ their visual ability to evaluate the significance of metadata as well as the characteristics of classification methods and distance functions. It can also be used to recognize and analyze patterns in large sets of images, and to get an overview of the entire body of pictures from a given set. The projection technique only uses a similarity function for calculating a suitable distribution of the points in the visualization space and has a linear time complexity.

Categories and Subject Descriptors (according to ACM CCS): H.3.3 [Information Systems]: Information Search and Retrieval—all subtopics Picture/Image Generation [I.3.3]: Viewing algorithms—Clustering [I.5.3]: –classification

1. Introduction

Our aim is the development of techniques which facilitate the search, exploration, and general structuring of large collections of works of visual arts. To this end we are searching for methods which can calculate and visualize relationships between pictures. The similarity between images in terms of human perception is obviously an important relationship between pictures, and there are projection techniques that place points representing individual data instances in a two-dimensional visualization space so that proximity in this space reflects some sort of similarity. These techniques support many analysis tasks for data sets. We use an algorithm that is well suited to calculate the similarity of pictures of visual arts in terms of human perception. However, this algorithm generates no vector representation reflecting similarity, which is the basis of most of the common projection techniques. Common methods are based on a given vector representation of the data or their time complexity is at least quadratic. Our proposed method has linear time complexity and is thus applicable to very large sets of images. Furthermore, it only requires a distance function for calculating the distribution of the points.

2. Similarity of Images

We use the algorithm given in [PHRE15] to calculate similarity between pictures. The algorithm is based on the assumption that fixations during the perception of visual arts,

along with their surroundings, constitute important image information both for recognizing and for comparing pictures (more information in [HS95], [Saa93], [JC80], and [HNA^{*}10] Chapter 3.2). The algorithm calculates a sequence of 100 fixations per image. The positions and the image information within a radius of 32 pixels of the simulated fixation points of two pictures are the basis for calculating 18 local comparison features (more information in [PHRE15] and [Aly11]).

The assumption underlying the method is that all features have an impact on how people perceive similarity, but that it is not known how strong this influence is. The method deals with the issue of choosing appropriate weighting factors by implicitly performing a weighting: To calculate the similarity between a picture P and a number of other pictures, the method first calculates all comparison features $x_{i,j}$ (i : feature number; j : picture id) between the picture P and the rest of all available pictures. Next, the mean value AM_i and the standard deviation σ_i is calculated for each of the comparison features. Now, the similarity between Picture P and another picture is regarded as the sum of normalized comparison factors \tilde{x}_i : $\tilde{x}_i = (x_i - AM_i)/\sigma_i$. This approach takes the range of the similarity features into account, and normalizes the variance of the distances between all objects and a single object to 1.

The evaluation of the similarity function in [PHRE15] revealed that the method is capable of identifying those pic-

tures in a set that are similar to a given picture in terms of human perception; nevertheless, not every picture that the calculations determined to be similar to a given picture was perceived as being similar by human viewers. As a distance function, the method has some drawbacks, because for every picture in a set of pictures the comparison factors are calculated/weighted individually: The function is not symmetric; the function does not obey the triangle inequality; in order to calculate the similarity of two pictures, it is necessary to compare one of the two pictures with all the pictures in the set. However, the projection technique presented in this paper is designed to handle these problems with linear time complexity.

3. The Projection Technique

For the intended projection technique there is only a distance function available and the technique should have linear time complexity. The methods commonly used are not suitable for this case: One class of similarity-based projection techniques are the methods of force-directed placement (see e.g. [Ead84], [Cha96], and [TMN03]). The fastest of these methods has a time complexity of $O(n^{5/4})$ [MC04] and would be fast enough for our purposes, but all methods with a time complexity faster than $O(n^2)$ work on the basis of a given distribution of the points in a multidimensional space, which is not available in our case. Another similarity-based projection technique is t-SNE [VdMH08], a variation of Stochastic Neighbor Embedding [HR02]. The technique keeps the low-dimensional representations of very similar data points closely together, which is advantageous for our purposes. Optimized versions have linearithmic time complexity [vdM14], but again, all methods with a time complexity faster than $O(n^2)$ work on the basis of a given distribution of the points in a multidimensional space. Another class of methods is focused on preserving cluster structures (e.g. in [CBP09] and [CLRP13]), which would be desirable for our purposes. However, they need either a given distribution of the points in a multidimensional space, or the distance of all pairs of points, which results in quadratic time complexity. Another drawback all procedures described above have in common is that these algorithms are iterative; thus, the results vary strongly with changing starting positions or if new objects are added. Therefore, the results are not stable.

The idea of our method consists of several techniques: assuming that the objects in question are represented by points in some unknown multi-dimensional space in which the Euclidian distance between the objects is proportional to the similarity of the objects; using principal component analysis (see e.g. [CC08]) to get the first two principal directions; building a plane space with these directions; and projecting the points into this space. The principal component analysis is a commonly used approach to reduce a multi-dimensional problem to a two-dimensional problem. As there is only a distance function available, and we want to avoid quadratic

time complexity, we have to create a suitable approximation solution.

- As an approximation for the first principal direction we construct a line AB through the points A and B with the greatest distance. To calculate the points with the greatest distance, n^2 distance calculations are necessary (n is the number of objects), so we calculate these points with an algorithm that, again, is only an approximation but has linear time complexity: We start with an arbitrary object and calculate the object with the greatest distance to this point. Then we once again calculate the point that is farthest from the object just calculated. We take the last two points as points A and B , which have approximate maximum distance to each other (see [OK89]).
- In order to project all objects O_i orthogonally to line AB , we calculate the similarity distances a_i , b_i , and c (see Figure 1). We take the line AB as our first dimension for the visualization space with A as the origin, and define $x_B = 1$. The coordinate value x_i can be calculated as follows: $x_i = a'_i/c$ where $a'_i = \sqrt{(a_i^2 + c^2 - b_i^2)/2c}$.
- As an approximation for the second principal direction we consider the line RS through the two points with the greatest distance, measured orthogonally to line AB . R is the point with the greatest orthogonal distance $h_i = \sqrt{a_i^2 - a'_i^2}$ to line AB (see Figure 1). The point with the greatest distance $r'_i = \sqrt{r_i^2 - (a'_i - a_i)^2}$ to R (see Figure 1) is taken as S .
- We take the line RS as our second dimension with R as the origin, and define $y_S = 1$. The coordinate value y_i for the second dimension is calculated analogously to the first dimension.

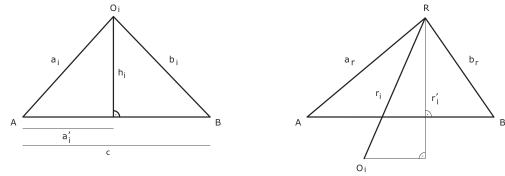


Figure 1: Geometric relationships.

The distance function does not obey the triangle inequality. It is possible that $c > a_i + b_i$, $a_i > c + b_i$, or $b_i > c + a_i$, the last two cases only in the second direction. In these cases the coordinate values cannot be calculated in the manner described. In order to get a suitable value for x_i in these cases, we increase or decrease the distances a_i and b_i with the factor k so that equation $c = ka_i + kb_i$ is satisfied, and can now calculate x_i with the values ka_i and kb_i . Having y_i we proceed analogously. The entire method has linear time complexity. In particular, the weights of the comparison factors must be determined only for the 5 objects A, B, R, S and the start object of the iteration that calculates A and B . The points A, B, R, S can be retained, if only a small number (compared

to the total number) of images are added or removed. In this case, the distribution of the points is maintained.

In order to evaluate our method we used a corpus of visual arts containing about 4,000 pictures. Large values in the calculation of image distances mean only that the corresponding image pairs are dissimilar, and distances with values above 0.5 (on a scale from 0 to 1) cannot be meaningfully distinguished. Therefore, we attach great importance to good correlation only for small distances. Thus, we calculated the sample correlation coefficient not only for all object pairs but also for pairs within a certain distance range. The sample correlation coefficient for all object pairs was 0.42; for pairs with a distance of less than 0.5 the coefficient was 0.54; and for pairs with a distance of less than 0.3 the coefficient was 0.63. The relative deviation of the Euclidean distance from the value of the distance function was 0.4 for all pairs, 0.33 for pairs with a distance of less than 0.5, and 0.3 for pairs with a distance of less than 0.3.

These values show that the presented point placement method presented here largely preserves the similarity between the objects. Together with a computation time of about one second for 4,000 pictures (we use an Intel i7-2600 processor with 3.4 GHz and 16 GB computer memory) and linear time complexity, the presented method is therefore well suited for the task at hand.

4. The Display

The display is divided into three linked views (Figure 2). The size of the views depends on the size of the top left visualization space, which is variable in size. The top right view shows those pictures which are in the light gray area of the visualization space. The picture displayed in the top left corner of this view, however, is the one next to the mouse cursor in the visualization space. The bottom view shows collected pictures, which can for example be stored or used to build up clusters. Detailed information about each image is available: A double-click on an image opens a window that displays information such as metadata, linked texts, or detail shots.

5. Visualization of Patterns

Metadata, such as the names of the artists, the pictures' titles, their years of origin, painting techniques, and sizes, help



Figure 2: The main Display.

to structure large amounts of images. The points in the visualization space can be colored according to user selected metadata thus showing their distributions (Figure 3). That way, the significance of the selected metadata can be easily understood. In contrast, when the significance of metadata is known, the characteristics of the similarity function that was used can be tested in this way. When a classification algorithm is used for the coloring of points, the visualization space shows the characteristics of the classification algorithm.

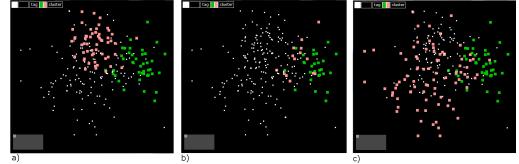


Figure 3: (a) Landscapes (green dots); portraits (red dots). (b) Pictures painted by Hockney before 2008 (green dots); painted after 2007 (red dots). (c) Pictures painted by Hockney (green dots); pictures painted by Rothko (red dots).

6. Reducing complexity

In large image collections, pictures are often featured several times with only small variations. Therefore, it seems only natural to visualize similar images only through one representative, which is why we included an agglomerative cluster method [KR08] to find those similar images:

- For each object pair, we apply the distance function in both directions and calculate the Euclidian distance of the images in the visualization space. Of the three distance values, we take the largest. The Euclidian distance must be adapted to the values of the distance function. We do this by multiplying it by a factor. This factor must be determined experimentally, which in our case was not problematic. We used the same factor for all sets of pictures. In contrast to the distance values of the distance function the Euclidean distance also takes the distance to all nearby objects into consideration, and, in all calculations, the results were better when the Euclidean distance was taken into account.
- We sort the object pairs according to their distance values.
- Starting with each object as the potential core of a cluster, we merge the clusters. To do so, we set the maximum distance value and progress through the ordered list of pairs of objects starting with the pair with the lowest distance value. We merge two clusters if the distance values of all pairs of the cluster members are less than the maximum value.
- We begin with a maximum distance of 0.01 and repeat the previous step while always increasing the maximum distance value by 0.01. We end the procedure once the maximum distance value has reached a specified limit.

We tested the cluster method on the whole corpus of 4,000 pictures as well as on a subset of 232 images. Applied to the small subset with the maximum distance set to 0.3, 44 clusters were created and they were chosen in a way that no subsequent manual improvement was necessary (Figure 4). If the maximum distance was set to 0.5, the number of clusters was cut in half, but in some clusters images were grouped together that did not fit well in terms of human perception. Applied to the set with 4,000 pictures with a maximum distance of 0.3, the results were still good, but not as good as in the previous case. The number of clusters was in this case about one third of the number of images, and could not be significantly reduced with manual rework.

The problem with the method presented here is that its time complexity is quadratic. The sorting has linear time complexity, because we do not have to sort precisely, but only have to assign the pairs to 100 distance classes (0.0 - 0.01 to 0.99 - 1.0). But we have to calculate the distance for every pair of pictures. The computation time for clustering the corpus of 4,000 images is about two minutes. Therefore, the method is only suitable for image sets with a size of several thousand pictures.



Figure 4: Some typical clusters calculated with our cluster method applied to the set of 232 pictures. Clusters are separated by larger spaces.

7. Conclusion

The evaluation shows that the presented projection technique preserves similarity to a high degree. Because it also has a linear time complexity, the technique is very suitable for the task at hand. The examples show that the similarity function used is well suited to identify similarities (e.g. the same artist, style, or subject matter). Nevertheless, we consider the provided method only as a first step towards an analysis tool for large image sets. The time complexity of the clustering method has to be improved without losing its current characteristics, so that the method can be applied to very large sets of images. However, our focus in the development of the tool is on the integration of the user into the analytical process. With visual analytics we want to include the user's knowledge and requirements in the design of the similarity

function, the clustering process, and in the detection and visualization of structures.

References

- [Aly11] ALY M.: *Searching Large-Scale Image Collections*. PhD thesis, California Institute of Technology, 2011. [1](#)
- [CBP09] CHOO J., BOHN S., PARK H.: Two-stage framework for visualization of clustered high dimensional data. In *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on* (2009), IEEE, pp. 67–74. [2](#)
- [CC08] COX M. A. A., COX T. F.: *Handbook of Data Visualization*. Springer Berlin Heidelberg, 2008. [2](#)
- [Chal96] CHALMERS M.: A linear iteration time layout algorithm for visualising high-dimensional data. In *Visualization '96. Proceedings*. (Oct 1996), pp. 127–131. [2](#)
- [CLRP13] CHOO J., LEE C., REDDY C. K., PARK H.: Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization. *Visualization and Computer Graphics, IEEE Transactions on* 19, 12 (2013), 1992–2001. [2](#)
- [Ead84] EADES P. A.: A heuristics for graph drawing. *Congressus Numerantium* 42 (1984), 146–160. [2](#)
- [HNA*10] HOLMQVIST K., NYSTRÖM M., ANDERSSON R., DEWHURST R., JARODZKA H., VAN DE WEIJER J.: *Eye tracking: A comprehensive guide to methods and measures*. Oxford University Press, 2010. [1](#)
- [HR02] HINTON G. E., ROWEIS S. T.: Stochastic neighbor embedding. In *Advances in neural information processing systems* (2002), pp. 833–840. [2](#)
- [HS95] HOFFMAN J., SUBRAMANIAM B.: The role of visual attention in saccadic eye movements. *Perception & Psychophysics* 57 (1995), 6: 787–795. [1](#)
- [JC80] JUST M. A., CARPENTER P. A.: A theory of reading: From eye fixations to comprehension. *Psychological Review* 4 (1980), 329 – 354. [1](#)
- [Kauf08] KAUFMAN L., ROUSSEEUW P. J.: *Finding Groups in Data; Introduction*. John Wiley & Sons, Inc., 2008, pp. 1–67. [3](#)
- [MC04] MORRISON A., CHALMERS M.: A pivot-based routine for improved parent-finding in hybrid mds. *Information Visualization* 3 (2004), 109–122. [2](#)
- [OK89] ÖMER, KALANTARI B.: Approximating the diameter of a set of points in the euclidean space. *Inf. Process. Lett.* 32, 4 (Sept. 1989), 205–211. [2](#)
- [PHRE15] PFLÜGER H., HÖFERLIN B., RASCHKE M., ERTL T.: Simulating fixations when looking at visual arts. *ACM Transactions on Applied Perception* 12, 3 (May 2015). [1](#)
- [Saa93] SAARINEN J.: Shifts of visual attention at fixation and away from fixation. *Vision Research* 33 (1993), 8: 1113–1117. [1](#)
- [TMN03] TEJADA E., MINGHM R., NONATO L. G.: On improved projection techniques to support visual exploration of multidimensional data sets. *Information Visualization* 2, 4 (Dec. 2003), 218–231. [2](#)
- [vdM14] VAN DER MAATEN L.: Accelerating t-sne using tree-based algorithms. *Journal of Machine Learning Research* 15 (2014), 3221–3245. [2](#)
- [VdMH08] VAN DER MAATEN L., HINTON G.: Visualizing data using t-sne. *Journal of Machine Learning Research* 9, 2579–2605 (2008), 85. [2](#)

Multiple Material Meshes for Erosion Simulation

V. Skorkovská^{1,2} and I. Kolingerová^{1,2}

¹Department of Computer Science and Engineering, Faculty of Applied Sciences, University of West Bohemia, Czech Republic
²NTIS - New Technologies for Information Society, University of West Bohemia, Czech Republic

Abstract

Triangle meshes have a very strong position in the computer graphics. They are getting more popular even in the field of erosion simulation, where the volumetric representation used to prevail. The real-life erosion scenes are usually formed of multiple materials and so a reliable means of material definition is needed. This paper proposes several easy-to-use approaches for multiple material definition, based on the space subdivision. Binary space partitions are used to simulate complex multi-material scenes. The approach allows the definition of a nontrivial scene composed of several materials, including the definition of gradually changing material.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Boundary representations

1. Introduction

Surface triangle meshes are becoming more and more popular in various fields of research, such as medical applications, architectural design or gaming industry. Material properties usually have to be assigned to the mesh to describe its visual appearance or to define its structure, which is necessary for applications working with the haptic feedback.

A multi-material scene is commonly represented as a set of non-intersecting meshes, one for each material present in the scene. This representation is efficient for static scenes with several separated materials, however, it may not be sufficient to describe a dynamic scene containing several materials blending into each other, e.g., during an erosion simulation. During such a dynamic simulation the surface mesh evolves and a more sophisticated way to describe the material properties is necessary.

We propose a material description approach suitable for the use in dynamic simulations. Our method uses binary space partitions (BSP) to define the different materials in the scene, with an optional definition of a distance function to represent a continuous change of the material. We simulate the erosion forces by the smoothed particle hydrodynamics (SPH) [GM77] particles in a way similar to the approach used by Krištof et al. [KBKS09], but any other erosion simulation could be easily used as well.

2. Related Work

Most of the work on domains with multiple materials focus on extracting a correct and consistent mesh for each of the materials present in the volumetric data obtained, e.g., from a medical scan. Wu and Sullivan [WS03] enhanced the marching cubes algorithm to reconstruct multiple material meshes. Zhang et al. [ZHB10] generate the mesh using an octree-based isocontouring method. Wang [Wan11] generates the mesh surfaces using a ray representation of a solid as an intermediate structure. These approaches are suitable for static scenes, where the domains are strictly separated.

For dynamic scenes or scenes, where the individual domains are blending into each other, the aforementioned approaches are inappropriate. An example of such a scene could be an erosion scenario, where sand and pebbles of various size mix up to form a river bank, that is being eroded by the flowing water. We could describe such a scene using a volumetric approach similar to the one used by Benes et al. in [BTHB06]. The volumetric representation is very memory consuming, a layered data representation introduced by Benes and Forsbach [BF01] can be used instead to alleviate the problem. The layered data structure is a sufficient description of a terrain scene consisting of several layers of material, but for a general scene with gradually changing material, it converges back to the volumetric representation.

A different approach is used by Tychonievich and Jones in [TJ10], where a Delaunay deformable model is used to

represent the eroded terrain and the material is defined for each cell of the Delaunay triangulation. A new mesh is generated every iteration of the method and the material properties have to be reconstructed using the resemblance of the two meshes.

The proposed solution represents the scene with a surface mesh for each individual object in the scene, regardless of the object material. The material is then assigned through a separate BSP structure, defining the material volume throughout the scene.

3. Multiple Material Meshes

A real-life scene is composed of objects made of different materials. Objects made of different materials are eroded in a different way; hard and resistant materials are eroded slowly, while the erosion of soft materials is happening much faster. To be able to simulate such phenomena, we need means to consistently describe the material of an object. A common way of representation is to have a separate mesh for each material present in the scene. This approach is suitable for simulation of static scenes, where the materials are strictly separated. If the scene contains objects with material gradually changing, a different approach is necessary.

We have tested several ways of material description that allow the definition of multiple materials for a single mesh, ranging from a material definition for each vertex of the mesh to a more sophisticated method of binary space partitions (BSP). We describe the methods in the following text.

3.1. Material in a Vertex

The most simple way to define multiple materials for a single mesh is to assign material properties to each individual vertex. This approach is very easy to implement, however, it brings many disadvantages as well. Storing material properties for each vertex takes up a lot of memory. Furthermore, this kind of material definition applies only to the surface of the object, not to the volume. Using this approach, the result of the erosion simulation will change according to the direction of the erosion. If the erosion direction is parallel with the boundary of the individual materials, the erosion will be simulated correctly. Figure 1 shows an object made of two different materials. The orange material is hard and sturdy, while the green material is soft and easily erodable. Figure 1b captures the result of the erosion simulation.

The disadvantage of the approach is insinuated in Figure 2. This time the boundary between the materials is perpendicular to the erosion direction. Figure 2b shows that the soft green vertices have been eroded past the boundary between the materials; assuming that the boundary was supposed to be straight. This is the main downside of the method, the fact, that we have no information about the volume of the material, we operate only with the surface.

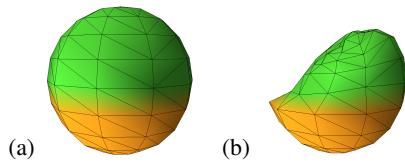


Figure 1: Material properties assigned to each vertex (a), erosion applied from the left gives correct results (b).

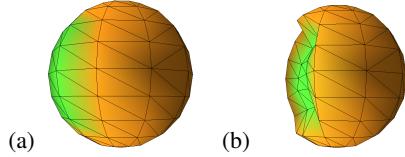


Figure 2: Material properties assigned to each vertex (a), erosion applied from the left gives incorrect results (b).

3.2. Division by a Plane

The problem of the previous approach can be reduced, if we define the material properties for the whole volume of the scene, using a plane to separate different materials. Figure 3 shows the situation from Figure 2, but the material definition is done by the dividing plane. Material is then assigned not to each vertex, but is dynamically determined based on the eroded vertex location during the simulation. The simulation gives the expected result, as shown in Figure 3b.

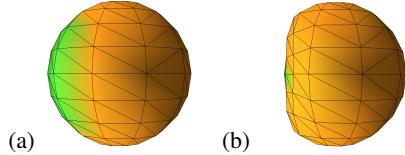


Figure 3: Material defined by a dividing plane (a), correctly eroded model (b).

This approach allows us to simulate a simple terrain composed of several materials, imitating the layering nature of the terrain. Figure 4 captures a simple terrain composed of two materials. The hydraulic erosion is simulated using the SPH particles, which erode the upper layer of soft material, exposing the underlying hard one (Figure 4b).

A similar approach is used in the work of Tychonievich and Jones [TJ10], where the authors define the toughness of the material as a function of the z-coordinate. They demonstrate the results on an example of water flowing through a rock canyon, eroding its soft sides and the bottom made of tougher material (Figure 5). We modeled a similar scene, the results are shown in Figure 6.

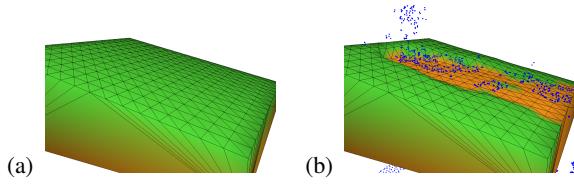


Figure 4: Material defined by a dividing plane (a). Hydraulic erosion is simulated by the SPH particles (b).

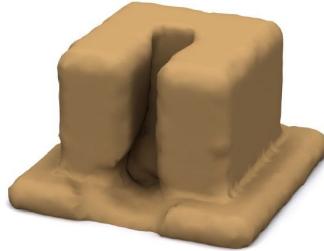


Figure 5: Water erodes the sides of the canyon [TJ10]

3.3. Division by a Function

The division by a plane can imitate the material distribution of a simple small-scale scene, however, in bigger scale, the boundary between materials will usually not be linear. To simulate a more complex boundary, a division function may be used. This approach allows us to describe a non-linear boundary analytically, without the use of a memory demanding data structure such as the volumetric approach. Figure 7 shows a demonstrative example of the function division approach. The terrain starts as a plane and is eroded by the water particles. The boundary between materials is defined by Equation 1 for Figure 7a and by Equation 2 for Figure 7b. The downside of this approach is the need to enumerate the function for each eroded vertex to determine the material. Also, it may be very difficult to describe a complicated boundary with a single division function.

$$z = \cos\left(\frac{x}{2} + \frac{y}{3}\right) - \sin\left(\frac{y}{3}\right) - \frac{x}{3} + 9; x, y, z \in \mathbb{R} \quad (1)$$

$$z = \sin\left(\sqrt{x^2 + y^2}\right) - \frac{x}{3} + 9; x, y, z \in \mathbb{R} \quad (2)$$

3.4. Binary Space Partitions

To simulate more complex distribution of the material, we are using binary space partitions (BSP). This approach extends the method described in Section 3.2 and allows us to split the scene up into more parts, using multiple division planes. The division planes are stored in a binary tree, each

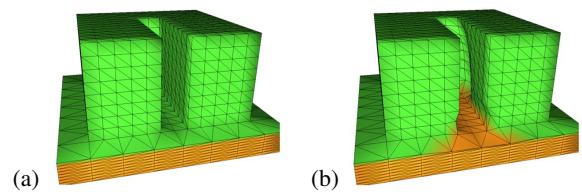


Figure 6: Canyon erosion. Original (a) and eroded mesh (b)

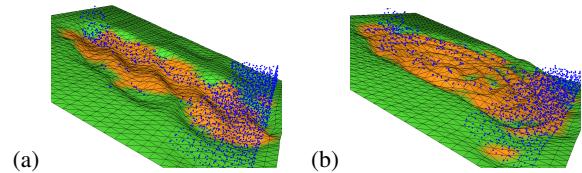


Figure 7: Division defined as Eq. 1 (a) and Eq. 2 (b)

leaf of the tree containing the information about the material of the subspace delimited by the corresponding planes. To decide which material should be assigned to a vertex, we need to search the BSP tree and determine for each level of the tree, on which side of the division plane the vertex lies. When we reach the leaf of the tree, we have found the subspace the vertex belongs to. The BSP approach is much more memory efficient than the commonly used volumetric representation, as the memory consumption depends on the number of blocks of different materials, not on the size or complexity of the mesh itself.

Figure 8 shows an example of BSP material definition. The scene has been divided into 16 spatial cells, each one has been assigned a random unique material. Darker color of the mesh marks the regions made of a tougher material; the lighter the color, the less durable the material is. As can be seen in Figure 8, the parts of the mesh made of less durable material are being eroded much faster than the the regions made of a tougher one.

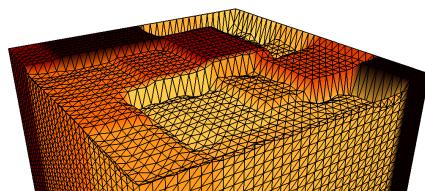


Figure 8: Material defined via BSP. Softer material gets eroded faster than the tougher one.

We can also extend this method to support the definition of a gradually changing material to be able to represent materials blending into each other, such as sand and pebbles at

a river bank. This is achieved by the addition of a distance function. The material properties are computed based on the distance of the eroded vertex from the division plane. An example can be seen in Figure 9. In this instance, the toughest material is assigned to the vertices on the division planes, as we move farther from the planes, the material gets softer.

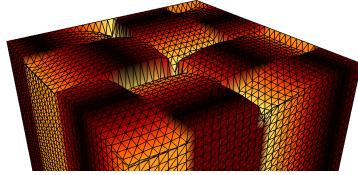


Figure 9: Material defined via BSP and a distance function.

The two aforementioned approaches can be combined as seen in Figure 10. Each spatial cell has been assigned a random unique material. With growing distance from the division plane, the material grows weaker. The division planes do not need to be axis-aligned, any plane can be used. An example of BSP using general planes and a distance function is shown in Figure 11.

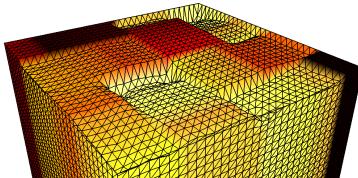


Figure 10: Material defined via BSP and a distance function with unique properties for each spatial cell.

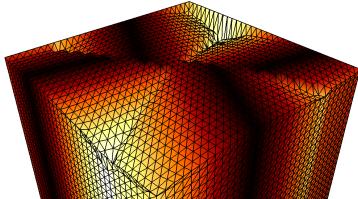


Figure 11: BSP material definition using general planes.

4. Results and Discussion

We have implemented and tested the discussed methods of multiple material definition for triangle meshes. We have confirmed the applicability of the methods for the use in simple erosion simulation scenarios, while having very low memory requirements. However, for big and complex scenes, the use of the proposed methods may be troublesome as the definition of the division planes by hand would be complicated and time-consuming.

5. Conclusion

The erosion simulation is a very important topic of computer graphics. More and more often, triangle meshes are used to represent the eroded terrain. Means to simulate the erosion of scenes composed of various materials are needed, but the commonly used approach of integrating the material properties with the vertices of the mesh may not be appropriate.

We have proposed a multiple material definition method based on the use of binary space partitions (BSP). The BSP tree subdivides the scene into cells and we define the material for each of them separately. The approach also supports the definition of gradually changing material through the use of a distance function.

Our solution is suitable for the use in scenarios composed of several types of material, where it is possible to define the splitting planes by hand. For very complex scenes, the BSP definition by hand would be tiresome. For future work, we would like to design an automatic function for the detection of the splitting planes.

Acknowledgements

This work has been supported by the European Regional Development Fund (ERDF) - project NTIS (New Technologies for Information Society), European Centre of Excellence, CZ.1.05/1.1.00/0.2.0090 and by the project SGS-2013-029 - Advanced Computing and Information Systems.

References

- [BF01] BENES B., FORSBACH R.: Layered data representation for visual simulation of terrain erosion. In *Computer Graphics, Spring Conference on, 2001*. (2001), IEEE, pp. 80–86. [1](#)
- [BTHB06] BENEŠ B., TĚŠÍNSKÝ V., HORNÝ J., BHATIA S. K.: Hydraulic erosion. *Computer Animation and Virtual Worlds* 17, 2 (2006), 99–108. [1](#)
- [GM77] GINGOLD R. A., MONAGHAN J. J.: Smoothed particle hydrodynamics - Theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society* 181 (Nov. 1977), 375–389. [1](#)
- [KBKS09] KRIŠTOF P., BENEŠ B., KŘIVÁNEK J., ŠŤAVA O.: Hydraulic erosion using smoothed particle hydrodynamics. *Computer Graphics Forum (Proceedings of Eurographics 2009)* 28, 2 (2009), 219–228. [1](#)
- [TJ10] TYCHONIEVICH L. A., JONES M.: Delaunay deformable mesh for the weathering and erosion of 3d terrain. *The Visual Computer* 26, 12 (2010), 1485–1495. [1, 2, 3](#)
- [Wan11] WANG C. C.: Computing on rays: A parallel approach for surface mesh modeling from multi-material volumetric data. *Computers in Industry* 62, 7 (2011), 660–671. [1](#)
- [WS03] WU Z., SULLIVAN J. M.: Multiple material marching cubes algorithm. *International Journal for Numerical Methods in Engineering* 58, 2 (2003), 189–207. [1](#)
- [ZHB10] ZHANG Y., HUGHES T. J., BAJAJ C. L.: An automatic 3d mesh generation method for domains with multiple materials. *Computer methods in applied mechanics and engineering* 199, 5 (2010), 405–415. [1](#)

Exact Bounding Spheres by Iterative Octant Scan

Thomas Larsson

School of Innovation, Design and Engineering
Mälardalen University, Sweden

Abstract

We propose an exact minimum bounding sphere algorithm for large point sets in low dimensions. It aims to reduce the number of required passes by retrieving a well-balanced set of outliers in each linear search through the input. The behaviour of the algorithm is mainly studied in the important three-dimensional case. The experimental evidence indicates that the convergence rate is superior compared to previous exact methods, which effectively results in up to three times as fast execution times. Furthermore, the run times are not far behind simple 2-pass constant approximation heuristics.

Categories and Subject Descriptors (according to ACM CCS): F.2.2 [Analysis of algorithms and problem complexity]: Nonnumerical Algorithms and Problems—Geometrical problems and computations; I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems

1. Introduction

The bounding sphere appears to be a ubiquitous tool for creating coarse conservative representations of more detailed and complicated geometric objects, such as polygon meshes and point clouds. The spheres are often arranged in a hierarchical data structure, the sphere tree, that provides a powerful representation of a complex object at various levels-of-detail [BO04]. Numerous applications in optimization, geometry, computer graphics, simulation, and games bear witness of the popularity of the bounding sphere. Not surprisingly, the optimal bounding sphere is known by many names, including minimum enclosing ball (MEB), minmax location, 1-center, and minimum covering sphere. To find it, we need to determine the point that minimizes the maximum distance to the input elements [EH72].

The advantages of using the sphere as a container stem from its simple shape, which gives a low storage cost and makes geometrical operations uncomplicated and fast. On the other hand, the main criticism concerns the quite poor fit it provides of the enclosed objects, which under certain difficult conditions makes the sphere inappropriate as a bounding volume [GLM96]. Even so, by combining the sphere with some other shape to provide a more flexible and finer approximate representation, much of the beneficial features of the sphere may still be exploited also in difficult scenarios. Such combinations have been utilized successfully to

speed up neighbour queries [KS97], ray tracing [WHG84], and collision detection [LAM09, CWK10].

Of course, to support bounding spheres, an appropriate construction algorithm is needed. It is straightforward to use a simple constant approximation method, which given n points in dimension d scans the input one or two times [ZZC06, Rii90]. However, since the main deficiency of the bounding sphere has to do with its size, any excessive space may be troublesome. The goal of this paper is to propose an algorithm that allows minimum bounding spheres to be computed rapidly enough for real-time computer graphics and simulation applications.

It is well-known that the MEB of a point set $P = \{p_0, p_1, \dots, p_{n-1}\} \subset \mathbb{R}^d$ is unique and defined by a support set of up to $d + 1$ points on its surface. Most previous exact methods target low-dimensional instances of the problem. A notable exception is the procedure given by Fischer et al. [FGK03]. Otherwise, computing the exact minimum sphere is doable in expected linear time in fixed dimension using Welzl's randomized algorithm [Wel91]. Although it has been demonstrated that this approach can be accelerated in practice by move-to-front and pivoting heuristics [Gär99], it may still require quite many passes over the input, and the computation of intermediate solutions may also involve solving numerous primitive cases to find valid support sets.

In the important three-dimensional case, we can use a more balanced way of retrieving *outliers*, i.e., points located outside the intermediate solutions, which is based on subdividing the space into octants during the linear scans of the input. In this way, the number of required passes can be reduced considerably, which in turns leads to fewer computations of intermediate solutions. Note that the same principle can be utilized more generally in dimensions beyond the third by subdividing the linear searches into hyper-octants, also known as orthants. However, their number grows exponentially with the dimension.

Based on these observations, we propose an exact MEB algorithm with an improved convergence rate and faster run time. Experimental validation confirms its efficiency and robustness in the three-dimensional case.

2. Algorithm

Conceptually, the algorithm is very simple. An initial minimum sphere is defined around a constant number of input points. The input is then scanned to find the farthest outlier in each octant using the current center as the origin for the search, which gives up to eight outliers in each pass. The tentative sphere is then updated to the next intermediate solution by taking these outliers in consideration. This procedure is repeated iteratively until no outliers can be found, meaning that the smallest possible sphere with a valid support set has been found. We refer to this approach as the *Iterative Octant Scan* algorithm.

This technique of incrementally enlarging the sphere and maintaining a valid support set follows G  rtner's approach [G  r99]. Since P is a finite set and the radius is monotonically increasing without ever exceeding the wanted minimum radius, the iterations can be repeated until the exact minimum sphere is obtained. In his approach, however, only "the most promising" point, i.e., the point with largest distance from the current center, is added in each iteration. By collecting a more complete view of the remaining outliers in each scan, the convergence rate can be improved. The following pseudocode describes the algorithm at a high level:

```

ITERATIVEOCTANTSCANMEB( $P$ )
1.  $S, c, r \leftarrow \text{SOLVESAMPLEDMEB}(P, m)$ 
2. repeat
3.    $Q, h \leftarrow \text{FARDESTOCTANTPNTS}(c, r, P)$ 
4.   if  $Q = \emptyset$  then exit loop
5.    $S \leftarrow S \cup Q$ 
6.    $S, c, r \leftarrow \text{SOLVEMEB}(S)$ 
7. return  $c, r$ 

```

The algorithm starts off by defining a MEB with center c and radius r around a constant number of points drawn randomly from the input set $P = \{p_0, p_1, \dots, p_{n-1}\}$. When n is large, we use the constant $m = 20$. The candidate support set S is also initialized with the supporting points of this initial



Figure 1: Illustration of the rapid progress of the iterative octant scan approach when applied on the Flowerpot mesh. The left image shows the initial sampled ball. Then the middle and right images show the intermediate solutions found in iteration 1 and 2, respectively. In iteration 3, a slightly larger ball is found (which looks just like the right image). Finally, iteration 4 simply detects that this is the final MEB.

MEB. Although it would be enough to base the initialization on a single input point, say by $c \leftarrow p_0$, $r \leftarrow 0$, and $S \leftarrow p_0$, the sampled MEB approach generally helps in reducing the number passes, without introducing much overhead.

Then the main loop is executed (Lines 2–6). On Line 3, a simple linear scan of the input is performed. The subroutine locates a set Q with up to eight extremal outliers, one from each octant, using c as the origin. Here extremal outlier refers to the input point that maximizes the Euclidean distance $\|p_i - c\| = \sqrt{(p_i - c) \cdot (p_i - c)}$ per octant. However, only points p_i satisfying $\|p_i - c\| > r$ are possible candidates for being extremal in any one of the octants. This condition is used to effectively filter the points, and only actual outliers are examined in more detail with respect to the octants they belong to. In fact, this linear scan is highly efficient with an almost negligible overhead compared to the standard approach of only finding the actual farthest outlier, which of course is always included among the octant points selected here.

Given that no outlier is found, the sought MEB has been found. This condition for termination is tested on Line 4. Otherwise, the set of outliers Q is added to the candidate support set S (Line 5), and the next intermediate solution is found by calling a subsidiary MEB solver that is able to handle the "low-level" primitive operations efficiently (Line 6). For more details, see G  rtner's treatment of how to robustly solve the primitive operations under floating point arithmetic [G  r99]. After this the main loop repeats until the exact solution is obtained. Figure 1 shows rendered images that illustrate the fast convergence of the algorithm in a specific case.

Clearly, S is a superset of the actual support set S' that defines the intermediate solutions. It is possible to reduce the size of S by assigning $S \leftarrow S'$ in the solver, which ensures $|S| \leq d + 1$ at the beginning of the next iteration. However, we have chosen to keep them around, since this can speed up the convergence for certain inputs, and the overhead of this appears to be insignificant for large n in 3D.

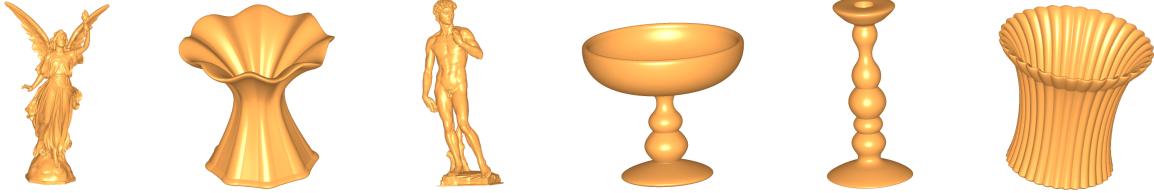


Figure 2: Some of the triangle meshes used to benchmark the algorithms.

Mesh	<i>n</i>	Octant Scan MEB			Farthest Point MEB			Ritter's method		
		<i>k</i>	<i>r</i>	<i>t</i> (ms)	<i>k</i>	<i>r</i>	<i>t</i> (ms)	<i>k</i>	<i>r</i>	<i>t</i> (ms)
Lucy	14,027,872	3	872.73962	80	6	872.73962	150	2	966.03467	60
Vase	4,609,442	4	13.769305	43	9	13.769305	78	2	16.842354	19
David	4,129,614	4	2713.4316	34	6	2713.4316	47	2	2842.9419	18
Goblet	1,008,772	4	15.644787	9.0	13	15.644787	24	2	18.267401	4.1
Blade	882,954	2	334.75421	3.8	3	334.75421	5.9	2	345.73364	3.8
Candlestick	456,482	3	37.304390	2.7	6	37.304390	5.0	2	39.363644	1.9
Flowerpot	410,482	4	26.173567	3.2	7	26.173563	5.0	2	32.095146	1.7
Hand	327,323	2	3.508905	1.2	4	3.508905	2.2	2	3.508966	1.4

Table 1: Minimum bounding sphere computation for n points in 3D. For each tested algorithm, the number of passes k , the resulting radius r , and the sequential run time t in milliseconds (ms) are listed.

Finally, we note that it is trivial to modify the termination criterion on Line 4 of the exact algorithm to create an alternative algorithm that reports $(1 + \varepsilon)$ -approximate solutions for any given $\varepsilon > 0$. The current farthest point distance h from c is already known by Line 3. Given that $h \leq (1 + \varepsilon)r$, the requested approximation is fulfilled and we can simply return the center c and radius h . This alternative may be used to speed up the computations, depending on the context.

3. Experiments

The presented algorithm was implemented in C++ and compiled in Visual Studio 2013. All tests were performed on a PC with a 2.80 GHz Intel Core i7-4810MQ CPU and 32 GB RAM under Windows 7. The input set was stored in a flat array of 3D points using single precision (32 bits) floating point numbers to represent the coordinates. The auxiliary MEB solver, however, used double precision to produce the intermediate solutions. All runs were executed single-threaded without vectorization. Note, however, that the dominating operation, the retrieval of the octant points with the largest distance from the current center, is susceptible of parallelization, an opportunity which we leave for future exploration (see some further comments about this in Section 4).

To challenge the proposed Octant Scan MEB method, we benchmarked it against two other well-known reference algorithms. The first competing strategy, called Farthest Point MEB, was basically Gärtner's algorithm [Gär99]. To make a fair comparison, however, we used our own adaptation, which shared as much source code as possible with the pro-

posed algorithm. In this way, it also benefited from the same effective initialization of the first intermediate solution based on sampling. In fact, the only difference is that it calls a standard farthest point routine instead of searching for the farthest octant points. The second competing strategy was Ritter's algorithm, which is extremely fast since it only uses two simple passes [Rit90]. The correctness is ensured by overestimating the radius using a greedy update strategy in the final pass (which is also used in [ZC06]).

A set of complex triangle meshes were chosen as benchmark problems. Several of the used meshes are well-known models from the Stanford 3D Scanning Repository and the Large Geometric Models Archive at Georgia Tech. The other used data sets will also be made publicly available. Rendered images of some of the included meshes are shown in Figure 2. We measured the number of passes k counted as the number of linear scans through the input points performed by the algorithm as well as the execution time t in milliseconds. The results are given in Table 1.

The experimental results indicate that the iterative octant scan method finishes in fewer passes than the competing exact method in each case. As shown by the Goblet model, the difference can be more than a factor of 3 (which led to a speedup of 2.7). The reported minimum radii were consistent between the two methods. Only very minor differences due to floating point rounding errors have been observed (as in the case of Flowerpot).

As expected, Ritter's approach, with its fixed two passes, gave a more predictable performance. In most cases, it had

the fastest execution, but for the Hand model our exact method was actually faster. Overall, the speedup for Ritter versus the exact octant scan method was in the range of 0.9–2.3. However, Ritter’s method overestimated the radius by a factor of 1.0–1.23 for the included test cases. This may be reasonable in certain applications, where exact solutions are not mandatory, but note that a radius increased by a factor of $\sqrt[3]{2} \approx 1.26$ doubles the volume in 3D.

To examine the correctness and robustness of our implementation, a brute force testing procedure was followed. Millions of random inputs with $n \in [10, 100]$ points in a cube were tested. In each case, it was confirmed that all points reside in the returned sphere (we used a small tolerance to account for minor rounding errors). The computed ball was also compared to the one given by Gartner’s publicly available miniball software. This testing procedure provoked no errors apart from very small differences in the returned radii apparently resulting from minor floating point rounding errors in the exact solver.

4. Conclusions

The proposed algorithm is able to find optimal bounding spheres of large point sets in just a few passes in the three-dimensional case (2–4 simple scans are often sufficient). Its usage of only k calls to a MEB solver on small subproblems to get the intermediate solutions further contributes to its high performance. To the best of our knowledge, this is the first exact algorithm with an execution speed that is almost comparable to that of naive constant approximation heuristics. In fact, when we ran the $(1 + \epsilon)$ -approximate version of our algorithm with $\epsilon = 10^{-2}$, it was faster than Ritter’s method in five out of the eight test cases, while it still produced smaller spheres in all eight test cases.

In the future, we would like to extend this paper by including more variants of the algorithm and by considering the behaviour more generally in low dimensions, say when $d \leq 10$. The same basic procedure can be applied, but in dimension d , there are 2^d hyper-octants to consider. Of course, this exponential growth seems problematic, but for large data sets, the approach may still be useful beyond the three-dimensional case. Additional filtering heuristics can also be designed to improve the performance, for instance by exploiting inherent spatial coherence in the data sets.

Furthermore, the possibilities for parallelizing the computations look compelling (cf. [KWL10, KL14]). The iterative octant scan method seems amenable for modern architectures supporting hierarchical levels of parallelism. By using a divide and conquer strategy, vectorized local scans can be performed on subsets of the input, and the results merged using a suitable reduction technique. In the end, this would hopefully lead to a fast and scalable data-parallel solution. This is definitely needed for massive data sets, such as scanned point clouds, which may contain billions of points.

Acknowledgements

The author is supported by the Swedish Foundation for Strategic Research through grant no. IIS11-0060. The models Lucy and David were kindly made available by the Stanford Computer Graphics Laboratory, the latter through the Stanford Digital Michelangelo Project Archive. The Blade and Hand models were provided by the Large Geometric Models Archive at Georgia Tech.

References

- [BO04] BRADSHAW G., O’SULLIVAN C.: Adaptive medial-axis approximation for sphere-tree construction. *ACM Transactions on Graphics* 23, 1 (Jan. 2004), 1–26. [1](#)
- [CWK10] CHANG J.-W., WANG W., KIM M.-S.: Efficient collision detection using a dual OBB-sphere bounding volume hierarchy. *Computer Aided Design* 42, 1 (2010), 50–57. [1](#)
- [EH72] ELZINGA J., HEARN D.: The minimum covering sphere problem. *Management Science* 19, 1 (1972), 96–104. [1](#)
- [FGK03] FISCHER K., GARTNER B., KUTZ M.: Fast smallest-enclosing-ball computation in high dimensions. In *In Proceedings of the 11th Annual European Symposium on Algorithms (ESA)* (2003), Springer-Verlag, pp. 630–641. [1](#)
- [Gar99] GARTNER B.: Fast and robust smallest enclosing balls. In *Proceedings of the 7th Annual European Symposium on Algorithms* (1999), Springer-Verlag, pp. 325–338. [1, 2, 3](#)
- [GLM96] GOTTSCHALK S., LIN M. C., MANOCHA D.: OBB-Tree: a hierarchical structure for rapid interference detection. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (1996), pp. 171–180. [1](#)
- [KL14] KALLBERG L., LARSSON T.: Accelerated computation of minimum enclosing balls by GPU parallelization and distance filtering. In *Proceedings of SIGRAD 2014* (June 2014), pp. 57–65. [4](#)
- [KS97] KATAYAMA N., SATOH S.: The SR-tree: an index structure for high-dimensional nearest neighbor queries. In *Proceedings of the 1997 ACM SIGMOD international conference on Management of data* (1997), ACM, pp. 369–380. [1](#)
- [KWL10] KARLSSON M., WINBERG O., LARSSON T.: Parallel construction of bounding volumes. In *Proceedings of SIGRAD 2010* (November 2010), pp. 65–69. [4](#)
- [LAM09] LARSSON T., AKENINE-MOLLER T.: Bounding volume hierarchies of slab cut balls. *Computer Graphics Forum* 28, 8 (2009), 2379–2395. [1](#)
- [Rit90] RITTER J.: An efficient bounding sphere. In *Graphics Gems*, Glassner A., (Ed.). Academic Press, 1990, pp. 301–303. [1, 3](#)
- [Wel91] WELZL E.: Smallest enclosing disks (balls and ellipsoids). In *New Results and New Trends in Computer Science*, Maurer H., (Ed.), vol. 555 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 1991, pp. 359–370. [1](#)
- [WHG84] WEGHORST H., HOOPER G., GREENBERG D. P.: Improved computational methods for ray tracing. *ACM Transactions on Graphics* 3, 1 (1984), 52–69. [1](#)
- [ZZC06] ZARRABI-ZADEH H., CHAN T. M.: A simple streaming algorithm for minimum enclosing balls. In *Proceedings of the 18th Canadian Conference on Computational Geometry* (2006), pp. 139–142. [1, 3](#)

Complete Quadtree Based Construction of Bounding Volume Hierarchies for Ray Tracing

Ulises Olivares^{†1} Arturo García² and Félix F. Ramos¹

¹ Departament of Electrical Engineering Center for Research and Advanced Studies of the National Polytechnic Institute, México

² Intel Corporation

Abstract

This paper presents an efficient space partitioning approach for building high quality Bounding Volume Hierarchies using x86 CPU architectures. Using this approach a structure can be built faster than a binned-SAH heuristic while the structure preserves its quality. This method consists of a hybrid implementation that uses binned-SAH for the top level and a binary partitioning approach for the rest of the levels. As a result, this method produces more regular axis-aligned bounding boxes (AABB) into a complete quadtree. Additionally, this approach takes advantage of the 4-wide vector units and exploits the SIMD extensions available for current CPU architectures. Using our construction approach a structure can be built up to three times faster than binned-SAH.

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.7]: Three-Dimensional Graphics and Realism—Raytracing

1. Introduction

Ray tracing is a rendering technique that produces high quality images. Nevertheless, it requires a high computational cost to produce a single image. This have generated special interest in using compute-intensive architectures such CPUs [WIK*06, SSK07], GPUs [LGS*09, ZHWG08] based implementations or even specialized architectures [NPP*11, DFM13]. Current improvements have been proved that is feasible to get interactive frame-rates even for complex scenes [Wal12, GMOR14, PFL*13].

However, although an acceleration structure decreases the complexity of ray tracing, it requires some time to be built, and this time depends on the quality of the construction and consequently on the heuristic that a structure uses. The use of sophisticated heuristics to estimate the cost of ray traversal tends to have a negative impact in the construction times but a positive impact in ray traversal performance, in an analogue way simpler algorithms tend to have faster construction times but a poorer ray traversal performance. The approach proposed in this paper tries to find a trade-off between complex split methods and fast split methods (get

faster construction times while it maintains the quality of the structure).

Currently, most of the research oriented to computer graphics uses GPU architectures to get lower bounds of execution times. Since it is possible to adapt complex structures to a linear representation [LGS*09], GPUs have been the compute-intensive architecture for excellence. Nonetheless, Wald et al. [WWB*14] have demonstrate that is possible to get competitive frame-rates by exploiting full compute capability of current x86 CPU architectures.

This paper proposes a method for building Bounding Volume Hierarchies (BVHs) for ray tracing based on a hybrid splitting approach. This approach uses binned-SAH for the top level in order organize primitives coherently and then it uses a binary split approach for the rest of the levels. This partitioning scheme produces a regular data structure into a complete quadtree, which ensures that all levels of the tree will be completed. This construction takes advantage of the 4-wide vector units and exploits the SIMD extensions available for current x86 CPU architectures. Additionally, this construction uses the integrated GPU on x86 architectures to perform compute-intensive operations efficiently.

[†] PhD Student of Computer Science at CINVESTAV México

2. Construction Overview

This section addresses a new approach for the construction of BVHs using a complete tree.

2.1. Complete Quadtree

In the construction approach presented by Garcia et al. [GMOR14] one of the main problems was the overhead produced by sorting primitives in each level of the tree and it produced lower construction times. In this paper, we proposed an approach that avoids this overhead by using arithmetic operations to select which primitives will be contained in the left or right internal sub-trees. This partition scheme is recursively applied until each level of the quadtree is filled. Finally, this partition approach produces a complete quadtree, this structure takes advantage of the 4-wide vector units of x86 CPU architectures.

In order to have a complete tree, it is necessary to determine how many primitives will be stored on every branch of the tree ensuring that only the deepest level could be incomplete.

Assuming that the number of leaf nodes N , is greater or equal than 4 (branching factor), where k represents the radix of the most significant bit of the binary representation of N , and r represents the complement $r = |N| - 2^k$ (Figure 1 depicts the binary representation of the number of primitives). Then, it can be two different cases: the binary representation of N is $10\dots_2$ or is $11\dots_2$. This binary representation follows a big endian order where the most significant bit is on the left side. In the first case, there are not enough leafs to fully complete the left side and thus the right side should be full, if $N = 2^k + r$ then we should have $2^{k-1} + r$ leafs on the left side and 2^{k-1} on the right side. In the second case, there are enough leafs to complete the left side and thus will have 2^k leafs while the right side will hold r leafs, for this case, $r \geq 2^{k-1}$.

The process just described provides the rules to determine the spatial partition (cut) in a given axis, it is represented by these two cases:

$$|N| = 2^k + r \quad (1)$$

$$|N| = \begin{cases} 10\dots_2, \\ 11\dots_2, \end{cases} \quad (2a)$$

$$|N| = \begin{cases} 11\dots_2, \end{cases} \quad (2b)$$

$$10\dots_2 = \begin{cases} 2^{k-1} + r, & \text{for } left \\ 2^{k-1}, & \text{for } right \end{cases} \quad (3a)$$

$$10\dots_2 = \begin{cases} 2^{k-1}, & \text{for } left \\ 2^{k-1} + r, & \text{for } right \end{cases} \quad (3b)$$

$$11\dots_2 = \begin{cases} 2^k, & \text{for } left \\ r, & \text{for } right \text{ with } r \geq 2^{k-1} \end{cases} \quad (4a)$$

$$11\dots_2 = \begin{cases} 2^k, & \text{for } left \\ r, & \text{for } right \text{ with } r \geq 2^{k-1} \end{cases} \quad (4b)$$

Lets assume there exist 13 primitives in a scene, so the number of primitives is given by $|N| = 2^k + r = 13$ from this formula we can obtain the most significant radix of the binary representation $2^k = 2^3 = 8$ and Finally, it is easy to obtain the r value from the original equation 1, $r = |N| - 2^k = 5$. Following the partition rules, we can calculate the number of primitives that will be contained in the left and right sides respectively.

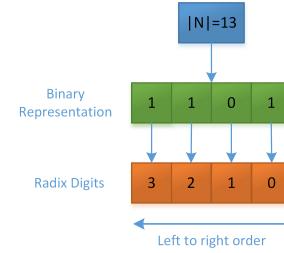


Figure 1: Binary representation of primitives to obtain 2^k , where $k = \text{radix digit}$.

2.2. Hybrid Construction

We employed a hybrid construction to take advantage of a complex split method that produce high quality Bounding Volume Hierarchies [Wal07] binned-SAH and a fast binary split approach that produces a complete tree just described above. We employed a binned-SAH on the top of the tree and the binary partitioning for the rest of the tree. The first partition ensures that coherent primitives are contained in the same bounding box and then, the binary split approach ensures that the tree structure will be a complete tree.

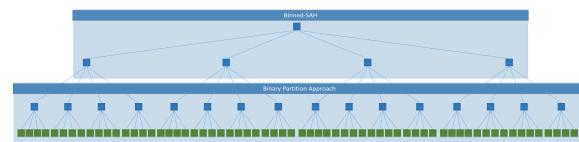


Figure 2: Hybrid Construction Approach. It uses binned-SAH for the top of the tree and our binary partitioning approach for the rest of the tree.

2.3. Space Partitioning

Once we have selected the number of primitives that every side will have, the next step is to decide which primitives will be stored in the left side and which will be stored in the right side. To do so, we find the lowest and highest values of the coordinates of the primitive centers using the axis that has the longest range, we select the primitives with the lowest value on that axis to be in the left side and the rest in the right side. Geometrically, we are selecting the longest side of the bounding box that contains all the primitive centers,

and splitting the primitives based on the order given by the projection of the centers on that side. Using this method will generate boxes that tend to be more regular, by reducing the length of the longest side.

Once the complete tree is built we know how many primitives will be contained in each node, then we need to perform the space partition in order to assign the primitives to each node. The space partitions are executed on the longest axis in order to improve the traversal performance by localizing the major density of adjacent primitives in the left branch of the tree.

The space partitioning is done following these steps:

1. Find the longest axis
2. Map Primitives
3. Split primitives

2.3.1. Find the Longest Axis

This is determined by doing a parallel reduction based on an algorithm presented by Hillis et al. [HS86] to sum an array of n numbers that can be computed in time $O(\log n)$ by organizing the addends at the leaves of a binary tree and performing the sums bottom-up at each level of the tree in parallel. Our function returns the longest axis by calculating the *max* and *min* values in x , y and z of an array of primitives, then the function calculates the distance between the *max* and *min* points for each axis and returns the axis that has the longest distance.

This compute-intensive operation was parallelized using Open CL and the integrated Graphics Processor Unit on x86 Intel current architectures (see Figure 3). Using the integrated GPU avoids communication latency which is implied when information is being transferred from the CPU to GPU using the PCI communication bus.

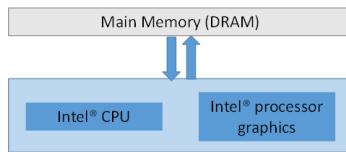


Figure 3: Shared memory on Intel architecture [Cor14].

2.3.2. Map Primitives

Garcia et al. [GMOR14] proposed a fast data parallel radix sort implementation to sort the primitives against the split axis in ascending order based on the centroids. Nonetheless, this sorting caused an overhead during the construction. As an alternative we proposed a simplest approach that map a primitive to a BVH node using arithmetic operations.

Assuming that it is necessary to take the first L primitives among N and the range of the values is $[a, c]$, it is selected a value to be the pivot that makes a cut proportional to L/N

and move all the primitives that are lower or equal to the pivot to the left and the rest to the right. As a result primitives are partitioned in two ranges $[a, b]$ and $(b, c]$. If L denotes the number of primitives contained on the first range, and L' denotes the number of primitives contained on the second range, then there exist two cases, this method guarantees that the left side is always complete:

$$L' = L \quad (5)$$

$$L > L' \quad (6)$$

In the first case, no further movements are needed, in the second case, it is necessary to update the lowest limit of our current range as the lowest value that we moved to the right and update L as $L - L'$ and N as $N - L'$. In the third case, b has to be updated as the highest value that we moved to the left and N has to be updated as L' . It is important to note that in every step we are removing at least one element from the array of primitives, this ensures that the algorithm will finish. Also, it can be observed that the complexity of the expected time will be similar to the expected cost for quickselect algorithms, which is $O(N)$ for every level.

2.3.3. Split Primitives

This step is straight forward because the split was calculated previously when the tree was pre-built to determine the number of primitives for each sibling node. Then, we apply the partition process described in the equations 2a and 2b, we get that the first primitives of the array will be assigned to the right node and the rest to the left node. After that, this rule is applied again in a recursive way until the four nodes in the quadtree are completed.

3. Results

This section exhibits the performance numbers of the Complete BVH structure rendering 3D models in a ray tracing application. The metrics taken compared the construction time and rendering frame rate against benchmarks obtained by State of the art acceleration structures on x86 architectures [WWB*14]. The tests were executed in a 2.5 GHz 8X Intel Core i7 CPU with 16Gb of DDR3 RAM compiled as 32-bit application using the ispc intel compiler.

The models used are Stanford Bunny (69,451 primitives), Crytek Sponza (279,163 primitives) and Happy Buddha (1,087,716 primitives). Figure 4 shows three-dimensional models rendered at 1024x1024 pixels. Tables 1 and 2 show that the proposed method offers lower construction times up to three times faster than binned-SAH. As a counterpart, binned-SAH construction has better ray traversal performance.

Table 1: Construction time for diverse scenes.

Scene	Hybrid Builder [Our Method]		Binned-SAH Builder Embree BVH4. [WWB*14]	
Ring(6 K)	2.829 ms	2.17 Mprims/s	7.840 ms	0.77 Mprims/s
Stanford Bunny(69 K)	10.850.48 ms	7.05 Mprims/s	33.083 ms	2.09 Mprims/s
Crytek Sponza(279 K)	43.657 ms	6.39 Mprims/s	179.598 ms	1.55 Mprims/s
Stanford Buddha(1 M)	174.170 ms	6.24 Mprims/s	403.810 ms	2.69 Mprims/s



Figure 4: Ring Model (6K), Stanford Bunny (69K), Crytek Sponza (279K) and Stanford Happy Buddha (1M) ray-traced using a Complete BVH4 acceleration structure at 1024 X 1024 pixels.

Table 2: Ray traversal performance for diverse scenes.

Scene	Hybrid Builder [Our Method]	Binned-SAH [WWB*14]
Ring(6 K)	154.075 fps	156.213 fps
Stanford Bunny(69 K)	120.523 fps	136.254 fps
Crytek Sponza(279 K)	5.03 fps	20.624 fps
Stanford Buddha(1 M)	2.01 fps	6.05 fps

4. Conclusions

This work presented a novel hybrid partitioning approach to build high quality BVH structure. It also presented an efficient process using SIMD extension to build the structure. The partition method took full advantage of current x86 CPU and integrated GPU architecture. We also presented very competitive build times of this construction. Ray traversal performance of this structure needs some improvements in order to increase the current frame-rate. The Complete BVH partitioning offers fast build times for high quality structures, efficient memory storage and provides high performance traversal for rigid objects. For future work we will apply this acceleration method in combination with predictive and adaptive strategies for accelerating the rendering of dynamic scenes.

Acknowledgments

The authors would like to thank the Stanford Computer Laboratory for the Happy Buddha and the Bunny models, Crytek for the Sponza model. Last but not least the CONACYT for its financial support.

References

- [Cor14] CORPORATION I.: Getting the most from opengl 1.2: How to increase performance by minimizing buffer copies on intel processor graphics, 2014. 3
- [DFM13] DOYLE M. J., FOWLER C., MANZKE M.: A hardware unit for fast sah-optimised bvh construction. *ACM Trans. Graph.* 32, 4 (July 2013), 139:1–139:10. 1
- [GMOR14] GARCÍA A., MURGUIA S., OLIVARES U., RAMOS F. F.: Fast parallel construction of stack-less complete lbvh trees with efficient bit-trail traversal for ray tracing. In *Proceedings of the 13th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry* (New York, NY, USA, 2014), VRCAI ’14, ACM, pp. 151–158. 1, 2, 3
- [HS86] HILLIS W. D., STEELE JR. G. L.: Data parallel algorithms. *Commun. ACM* 29, 12 (1986), 1170–1183. 3
- [LGS*09] LAUTERBACH C., GARLAND M., SENGUPTA S., LUEBKE D., MANOCHA D.: Fast bvh construction on gpus. In *In Proceedings of the EUROGRAPHICS ’09* (2009). 1
- [NPP*11] NAH J.-H., PARK J.-S., PARK C., KIM J.-W., JUNG Y.-H., PARK W.-C., HAN T.-D.: T&tengine: Traversal and intersection engine for hardware accelerated ray tracing. In *Proceedings of the 2011 SIGGRAPH Asia Conference* (New York, NY, USA, 2011), SA ’11, ACM, pp. 160:1–160:10.
- [PFL*13] PARKER S. G., FRIEDRICH H., LUEBKE D., MORLEY K., BIGLER J., HOBEROCK J., MCALLISTER D., ROBINSON A., DIETRICH A., HUMPHREYS G., MCGUIRE M., STICH M.: Gpu ray tracing. *Commun. ACM* 56, 5 (May 2013), 93–101. 1
- [SSK07] SHEVTSOV M., SOUPIKOV A., KAPUSTIN A.: Highly parallel fast kd-tree construction for interactive ray tracing of dynamic scenes. *Comput. Graph. Forum* 26, 3 (2007), 395–404. 1
- [Wal07] WALD I.: On fast construction of sah-based bounding volume hierarchies. In *Proceedings of the 2007 IEEE Symposium on Interactive Ray Tracing* (Washington, DC, USA, 2007), RT ’07, IEEE Computer Society, pp. 33–40. 2
- [Wal12] WALD I.: Fast construction of sah bvhs on the intel many integrated core (mic) architecture. *IEEE Trans. Vis. Comput. Graph.* (2012), 47–57. 1
- [WIK*06] WALD I., IZE T., KENSLER A., KNOLL A., PARKER S. G.: Ray tracing animated scenes using coherent grid traversal. *ACM Trans. Graph.* 25, 3 (July 2006), 485–493. 1
- [WWB*14] WALD I., WOOP S., BENTHIN C., JOHNSON G. S., ERNST M.: Embree: A kernel framework for efficient cpu ray tracing. *ACM Trans. Graph.* 33, 4 (July 2014), 143:1–143:8. 1, 3, 4
- [ZHWG08] ZHOU K., HOU Q., WANG R., GUO B.: Real-time kd-tree construction on graphics hardware. *ACM Trans. Graph.* 27, 5 (Dec. 2008), 126:1–126:11. 1

Visualizing Single-Camera Reprojection Errors Using Diffusion

Z. Franjcic^{1,2} and M. Fjeld²

¹Qualisys AB, Gothenburg, Sweden
²t2i Lab, Chalmers University of Technology, Sweden

Abstract

This paper reports on early-stage research into using dynamic scale-space representation of image point reprojection error data obtained during calibration of a single camera. In particular, we employ time-dependent simulation of the heat equation, diffusing the point reprojection errors over the entire image plane. Initial experiments show the expected effect of an originally large number of point reprojection error measurements being coalesced into a smaller number of relatively larger regions. We round off the paper by presenting ongoing work aiming to exploit the time-progression of simulations to convey further information and thereby assisting manual visual analysis.

Categories and Subject Descriptors (according to ACM CCS): I.4.1 [Image Processing and Computer Vision]: Digitization and Image Capture—Camera calibration

1. Introduction

Non-photogrammetric and semi-photogrammetric camera calibration procedures in computer vision often aim to minimize a reprojection error [Zha00]. This type of error describes how well the camera projection model captures the mapping from object points in space to points on the image captured by the camera. Visualizing the reprojection error measure in an intuitive way is often not easy, especially if many object points have been acquired by the camera. Using only the spatial domain for visualization, that is, basing the visualization on the two-dimensional image plane, and possibly extending the representation into the third dimension, can lead to confusing graphical representations of the error.

In this early-stage work we aim to investigate the addition of a temporal dimension to two dimensional spatial visualization. Although not physically meaningful for a cumulative view of the point data set, we can still formulate a time-dependent transformation that leads to a scale-space interpretation [Lin93] of the data. Similar approaches have been examined, for example, in medical image processing to highlight hotspots in point-like images at progressive zoom levels [MDWL14]. However, here we do not consider zooming, but rather view the series of transformed images as snapshots of a time series. Such time series data may be viewed

as three-dimensional data sets with two spatial and one temporal dimension.

2. Reprojection error visualization

Given reprojection errors for a set of image points captured by a single camera, the task of manually finding patterns in such a collection can be very challenging in terms of human effort required, especially if the set contains thousands of image points (Figure 1). Here, we investigate alternative

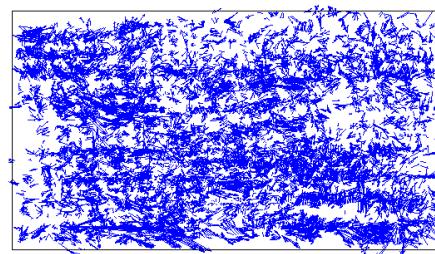


Figure 1: Deviations of measured image points from the image points predicted by a projection model for several thousand points accumulated from a few hundred images (frames).

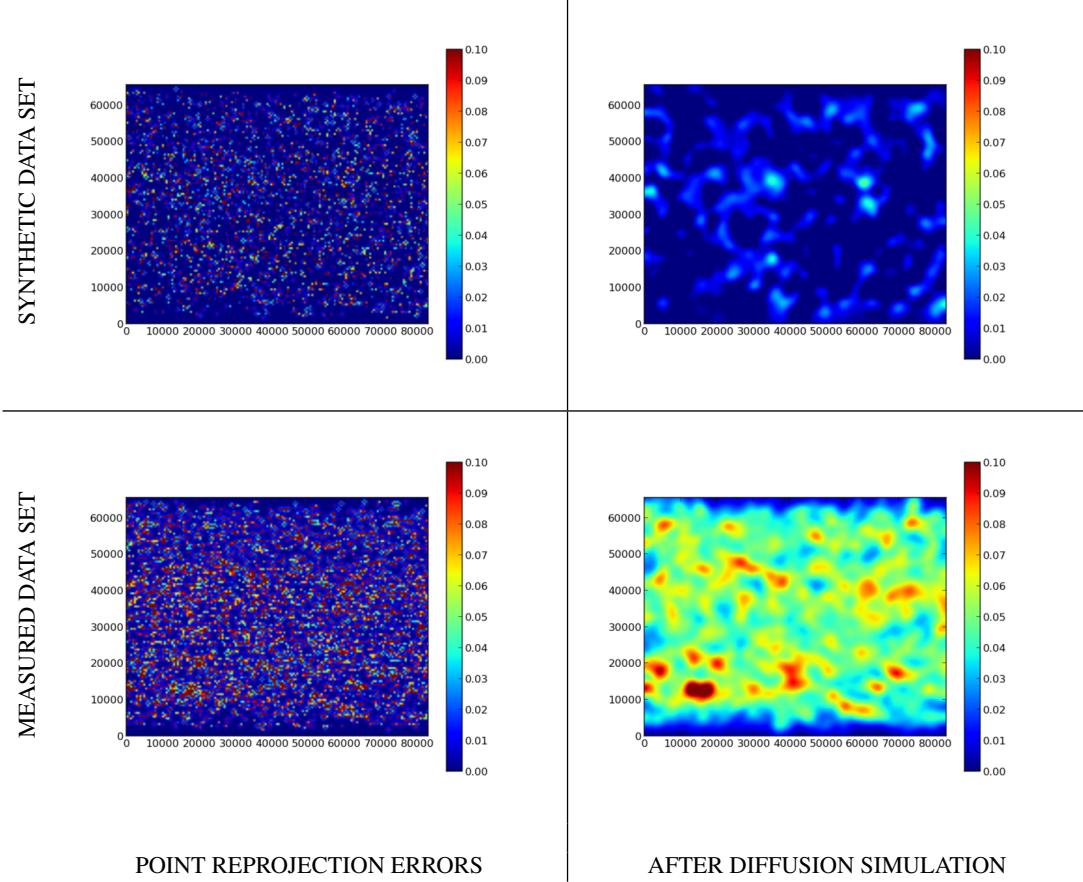


Figure 2: Heat equation simulation in the image plane for synthetic (top row) and measured data (bottom row). We apply diffusion simulation transforming the source images (left column) into filtered images (right column) to assist manual pattern recognition. The x and y axes represent the coordinates in the image plane in number of sub-pixels relative to the bottom left corner of the plane.

techniques that can help reducing the level of complexity introduced by larger data sets, such as clustering [JMF99] or principal component analysis [Jol05]. Once the large data set has been filtered or otherwise compressed, visualization of the processed data can render manual recognition of patterns more feasible. Heatmaps are a widespread visualization method [WF09]; we choose to focus on this technique for our early-stage research. We opt for ignoring directionality of reprojection error measurements and solely consider the magnitude of the errors.

3. Diffusion simulation

Motivated by results from scale-space theory [Lin93] we decided to explore diffusion-based simulation and visualization of the reprojection error magnitudes. Specifically, we chose a simple form of diffusion simulation based on the heat equation. The two-dimensional heat equation is given

by

$$\frac{\partial e(x,y,t)}{\partial t} = D \nabla^2 e(x,y,t) \quad (1)$$

where e is the temperature at location (x,y) at time t , D is the thermal diffusivity, and ∇^2 is the Laplace operator. In our case, e represents the reprojection error magnitude, and the location (x,y) corresponds to the pixel coordinates in the image plane. By simulating the heat equation, we obtain a multi-scale representation, with finer details damped with time [Lin93]. Hence, features that span larger areas of the image plane can potentially be revealed more easily than in the more detailed original data.

4. Preliminary results

We captured several thousand image points of a planar pattern and their reprojection error by performing calibration with a single camera [Zha00]. Figure 1 depicts the calculated

point reprojection errors as deviations of the measured image points from the points predicted by the calibrated projection model. We then simulated the heat equation according to Equation (1) setting the magnitude of the calculated point reprojection errors as initial point temperature values. In order to compare the simulation results against a baseline, we ran a simulation on artificial data. For this purpose, we generated synthetic reprojection errors around each image point by sampling random values from a standard normal distribution with adjusted standard deviation, where standard deviation was set to the maximum reprojection error magnitude obtained from the calibration. We then ran a simulation on the resulting artificial reprojection errors.

Figure 2 shows the start (left column) and end states (right column) of diffusion simulation performed for both the synthetic and the calibrated reprojection error data[†]. For the synthetic data the visualization of the simulation indicates a rather fast convergence to an equilibrium (top row). For the data from the calibration measurements the convergence is less pronounced after the same simulated elapsed time (bottom row), with larger and less evenly distributed coalesced errors. Based on this phenomenon of fusion of point errors into smoother, more visible error spots, we conjecture that diffusion-based simulation could possibly facilitate efficient visual identification of systematic reprojection errors.

5. Ongoing work

As a next step of this work, we plan to examine the effects temporal simulation may have on visualization. A possible effect could be that localizing the exact points contributing to an “error hotspot” formation as the simulation progresses in time is facilitated. Another effect could be that simulation allows for identification of similar patterns within and across reprojection error data sets based, for instance, on concepts related to heat kernel signatures [BK10]. Furthermore, we intend to conduct a user study to assess the usefulness of visualizing reprojection errors through diffusion simulations.

Specifically, we aim to investigate whether such visualizations actually could help a user in detecting patterns in the point reprojection error data set not easily recognized using traditional visualizations.

6. Acknowledgments

This work was supported by the EU FP7 People Programme (Marie Curie Actions) under REA Grant Agreements 289404 and 290227. The authors would to thank the anonymous reviewers for their helpful suggestions and comments.

[†] A video of the simulation together with a link to this paper can be found at <http://t2i.se/publications/>.

References

- [BK10] BRONSTEIN M. M., KOKKINOS I.: Scale-invariant heat kernel signatures for non-rigid shape recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on* (2010), IEEE, pp. 1704–1711. 3
- [JMF99] JAIN A. K., MURTY M. N., FLYNN P. J.: Data clustering: a review. *ACM computing surveys (CSUR)* 31, 3 (1999), 264–323. 2
- [Jol05] JOLLIFFE I.: *Principal component analysis*. Wiley Online Library, 2005. 2
- [Lin93] LINDEBERG T.: *Scale-space theory in computer vision*. Springer, 1993. 1, 2
- [MDWL14] MOLIN J., DEVAN K. S., WÄRDELL K., LUNDSTRÖM C.: Feature-enhancing zoom to facilitate ki-67 hot spot detection. In *SPIE Medical Imaging* (2014), International Society for Optics and Photonics, pp. 90410W–90410W. 1
- [WF09] WILKINSON L., FRIENDLY M.: The history of the cluster heat map. *The American Statistician* 63, 2 (2009). 2
- [Zha00] ZHANG Z.: A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22, 11 (2000), 1330–1334. 1, 2

Civic Participation and Empowerment through Visualization

Samuel Bohman[†]

Department of Computer and Systems Sciences, Stockholm University, Sweden

Abstract

This article elaborates on the use of data visualization to promote civic participation and democratic engagement. The power and potential of data visualization is examined through a brief historical overview and four interconnected themes that provide new opportunities for electronic participation research: data storytelling, infographics, data physicalization, and quantified self. The goal is to call attention to this space and encourage a larger community of researchers to explore the possibilities that data visualization can bring.

Categories and Subject Descriptors (according to ACM CCS): H.5.2 [Information Interfaces and Presentation]: Screen design—User-centered design

1. Introduction

Since the advent of the web in the early 1990s, prospects of electronic democracy have been viewed as heralding in a new era of political participation and civic engagement. However, empirical studies suggest most initiatives to date have failed to live up to expectations despite large investments in research. For instance, Chadwick [Cha08] states that “the reality of online deliberation, whether judged in terms of quantity, its quality, or its impact on political behaviour and policy outcomes, is far removed from the ideals set out in the early to mid-1990s”. In order to provide a future direction for electronic democratic participation, scholars have attempted to systematize current research and identify its main constraints and challenges. Macintosh, Coleman, and Schneeberger [MCS09] identified six main research challenges, barriers, and needs: fragmentation of research, immature research methods and designs, technology design, institutional resistance, equity, and theory. Other scholars have suggested that the problem with electronic democracy goes deeper. For example, Coleman and Moss [CM12] argue that the assumed deliberative citizen is a construction driven by researchers’ effort to produce responsible, democratically reflexive citizens modeled on Habermas’ discursive ideal of deliberative democracy.

The authors acknowledge that the deliberative approach to civic dialogue is “unduly restrictive, discounting other important ways of making, receiving, and contesting public claims”. They therefore encourage researchers in the field to be more open to a wider range of practices and technologies and suggest some of the most innovative research is being done in the area of computer-supported argument mapping and visualization. However, the visualization research they are referring to have primarily focused on facilitating large-scale online deliberation within a conventional rationalistic framework. In general, visualization has been an underused technology in electronic participation research [Boh14].

2. The Power of Data Visualization

Whereas the conventional deliberative approach to broaden democratic engagement and participation using the Internet has had limited impact beyond academic circles, the past decade has seen a strong growth in the use of data visualization to reach a wide target audience. A prime example is Al Gore’s narrated charts in the documentary film *An Inconvenient Truth* that earned him and the Intergovernmental Panel on Climate Change (IPCC) the Nobel Peace Prize in 2007. Another example is the approach developed by Hans Rosling, professor at Karolinska Institutet, Stockholm, Sweden, and co-founder of the Gapminder Foundation. Using humor and drama of a sportscaster and a piece of software that turns seemingly dry data into colorful animated graphics, Rosling debunks myths about the developing world while making it an enjoyable experience. These

[†] This research has received funding from The Swedish Research Council for Environment, Agricultural Sciences and Spatial Planning (FORMAS) under grant agreement no 2011-3313-20412-31.

observations, albeit of a few examples, suggest data visualization has potential for reaching out to a broad audience in innovative ways and facilitate greater citizen engagement in public affairs.

2.1. A Brief Historical Overview

To the casual observer, it would appear that data visualization is a recent phenomenon. In fact, the graphic portrayal of quantitative information has a long and rich history. According to Friendly [Fri08a], the beginnings of modern statistical graphics can be found in scientific discoveries, technological advancements, and societal developments in the 17th and 18th century. Many visualization techniques that are still being used today were introduced during this period including the line chart, bar chart, and pie chart.

In the second half of the 19th century, a number of developments combined to produce a “Golden Age of Statistical Graphics” [Fri08b]. Two well-known and much-discussed graphical exemplars from this period include John Snow’s dot map of a cholera outbreak in London and Florence Nightingale’s polar area charts displaying mortality rates of British soldiers during the Crimean War. Snow and Nightingale used their charts successfully as critical evidentiary statements in campaigning for improved sanitation that eventually led to government healthcare reforms.

In the first half of the 20th century, earlier enthusiasm for statistical graphics was supplanted by the rise of mathematical statistics [Fri08a]. Few graphical innovations were introduced during this period; it was, however, a time for consolidation and popularization. An important factor for the diffusion was pictorial statistics or pictograms. An influential advocate of pictograms was the Austrian philosopher Otto Neurath who developed a visual language known as Isotype with the purpose of explaining societal developments to the broad uneducated public. Due to the association of Isotype with left-wing movements and Soviet propaganda, the method disappeared during the Cold War in the Western world and its legacy has gone either unnoticed or unappreciated [Jan09]. In the last decade, however, the method has received renewed interest and increased attention. For example, Mayr and Schreder [MS14] review Isotype with respect to its potential for today’s civic education and participation and propose that we should rediscover its core principles and adapt them to a modern context.

The period from 1950 to 1975 constituted a rebirth for data visualization. Friendly [Fri08a] lists three significant events that contributed to the upswing. First, the publication of John Tukey’s book Exploratory Data Analysis [Tuk77]; second, the publication of Jacques Bertin’s work Sémiologie graphique [Ber67]; and third, the introduction of computers and software for statistical graphics. In 1983, Edward Tufte published the classic The Visual Display of Quantitative Information [Tuf83] which introduced several important

concepts, including “data-ink ratio”. At the turn of the century, Card, Mackinlay, and Shneiderman [CMS99] published a collection of seminal papers which established information visualization as a distinct field separate from scientific visualization.

As previously suggested, data visualization has exploded in the last ten years. Along with an increasing body of literature, the past decade has seen the emergence of a vast array of programming languages, toolkits, and libraries for interactive data visualization. Prefuse, for example, was an early visualization framework using the Java programming language [HCL05]. Today, many web-based data visualizations are built with D3, WebGL, and other JavaScript frameworks that utilize web standards and do not require web browser plug-ins.

3. Democratizing Data Visualization

In the following subsections, we explore four interconnected themes that contribute to a democratization of data visualization. The themes should not be viewed as comprehensive, but as stimuli to the research community to begin to ask better questions regarding the design of future technologies and practices for public participation.

3.1. Data Storytelling

Up to now, data visualization beyond the simple pie or bar chart has been the domain of specialists trained in either statistics or computer science. This was particularly true if dynamic and interactive charts and diagrams were required. However, the last few years have seen the emergence of a new class of self-service applications that support dynamic data querying, visual analysis, and interactive presentation on standard personal computers. These user-friendly data exploration and visualization tools, which are typically available free, empower the average user as it makes them less dependent on technical expertise and enable a broad audience to tell stories with data using visualization [KM13]. An example of an interactive data-driven story using visualization is the New York Times’ dialect quiz How Y’all, Youse, and You Guys Talk [KA13]. By responding to 25 different questions about the language the user most likely uses in different situations, the quiz builds a profile of the user’s dialect. When all questions are completed, a heat map indicates where in the United States the user most likely would find a person who uses a similar dialect. The dialect quiz became a huge success; in just eleven days (it was published on December 21) it became the most visited content of 2013 throughout www.nytimes.com, their mobile site and iOS apps [New14].

3.2. Infographics

Information graphics, a popular form of visualization commonly found in news media and often referred to as info-

graphics, has exploded in the digital age. The typical online infographic is a static high-resolution graphic design that attempts to transform abstract information about a specific topic into a format that is visually engaging, easily understood, and easily shared. Infographics combine data with design—numbers, data displays, words, and pictures—in order to inform, entertain, or persuade their audience. Despite the proliferation of infographics in today's fast-paced digital society, little research has been conducted on them. However, they typically share a number of common attributes. Similar to the idea behind micro-blogging services such as Twitter, the main characteristic of infographics is that their purpose is to tell the gist of a story at a glance. They are stand-alone visuals that are easy to digest and do not require additional information to be comprehensible. A second important characteristic is that infographics are meant to be aesthetically pleasing. Indeed, well-executed infographics are often admired for their beauty or for bringing out the beauty in data. A third characteristic of online infographics is that they are viral, easily shared and spread across social networks from person to person through “word of mouse”.

Although infographics are generally considered effective for disseminating information to the masses, statisticians and others have criticized them for relying too much on style over substance [Cai15]. Clearly, many infographic works are nothing more than eye candy that publishers and marketers use to gloss up their content, overly designed and convey little meaning. A surprisingly large number of infographics deceive their viewers by cherry picking statistics, warping facts, or providing questionable, vague, or nonexistent data sources [Kru13]. Despite these ethical objections, the sharp increase of online infographics in the last five years suggests that they appeal to a broad audience, a fact that makes them worth investigating further.

3.3. Data Physicalization

Physical data visualization, a lesser-known subfield of data visualization, studies alternative data representations where data is not represented through pixels on a computer screen, but via physical modalities experienced directly through the eye (not including ink on paper) or other human senses. In contrast to conventional data visualization where objectivity is the norm, physical representations of data allow for, sometimes even encourage, the inclusion of subjectivity in order to be evocative and increase onlookers' engagement. In an overview paper, Vande Moere [Moe08] explores the design space of physical data visualization in non-professional contexts. He lists five genres: data sculptures, ambient displays, pixel sculptures, object augmentation, wearable visualization, and alternative modalities. Data sculptures are data-driven physical artifacts that can be touched and explored through a tangible user interface. Ambient displays turn architectural spaces into a data display through subtle changes in light, sound, movement, solids, liquids, or gases

that can be processed in the back-ground of awareness. Envisioned as being all around us, ambient displays blur the boundary between the physical and digital worlds to create an interface between people and digital information. Pixel sculptures use non-screen-based visual units for representing information. An example is synchronized mass games or gymnastics, often seen at the Olympic Games, where each individual make up an element in a giant mosaic picture. Object augmentation refers to superimposing everyday objects with information. Visual animated projections on building facades and sidewalks is a common example. Wearable visualization draws on miniature computing devices that fits in clothing, jewelry, and other wearable things. The last category, alternative modalities, utilize non-visual representations of data that can be experienced through sound (sonification or auditory displays), touch (tactile or haptic displays), smell (olfactory displays), or taste (palatable interfaces). For instance, an experimental workshop called Data Cuisine explored food using culinary means as an alternative medium for representing data [Ste12].

3.4. Quantified Self

Self-knowledge through numbers, the motto of the quantified self grassroots movement [WK15], is quickly becoming a mainstream phenomenon. Currently, it is estimated that one in five U.S. adults are tracking their physical activity, sleep pattern, nutritional intake, and many other things related to their lives through a portable or wearable computing device such as a smartphone, smartwatch, or activity tracker [FD12]. The basic idea is simple: through more granular around-the-clock quantified monitoring, people can make smarter lifestyle choices and live healthier, more active lives. The reasons for self-surveillance are numerous and varied and range from the causal fitness-tracker who monitors his or her own exercise, to tech-savvy patients and citizen scientists who share their medical and lifestyle data online to help others and advance research, to life-logging enthusiasts with a passion for self-discovery through personal analytics. The story of Doug Kanter, who blogs about living with diabetes at <http://www.databetic.com>, offers a glimpse of what the future of personal data quantification might look like. In 2012, he used a suit of medical devices, activity trackers, smartphone applications, and PC software to record all his diabetes data and physical activities. He visualized his yearlong quantified self project as a poster displaying every blood sugar reading, every insulin dose and meal, as well as all activity data. Kanter's systematic self-tracking approach helped him become more aware of his behavior and provided an opportunity for change. As a result, his diabetic control improved considerably, making 2012 the healthiest year of his life.

4. Conclusions

In this article, we have explored the potential of using data visualization to promote civic engagement. First, we reviewed some of the challenges of contemporary electronic participation research and found that it cautions us to reduce our expectations of the conventional approach to online participation since there is little evidence of its success. In particular, it suggests the rationalistic model based on deliberative theory has become a straightjacket, impeding wide civic involvement. This predicament has prompted some scholars to rethink their earlier views and suggest the study of online participation should be expanded to incorporate a wider range of technologies and practices. As a response to this call, we examined the power of data visualization through a couple of recent examples and a brief historical overview. We then explored four overlapping themes that suggest data visualization represents a yet untapped potential in promoting a more informed and engaged participation in civic and democratic life. The themes and examples discussed is suggestive (but by no means conclusive) evidence that the time is ripe for scholars to consider the use of data visualization in political participation and civic engagement research. However, the versatility and potential applications of data visualization in the service of democracy remain to be explored. I would like, therefore, to invite my colleagues to join me in exploring and reflecting on the following research questions:

- Techniques of storytelling focus on people, motives and contexts rather than numbers. How can stories help bring data to life?
- Aesthetics reaches us on a different level than words and numbers alone. How can data be combined with art and design to evoke emotional engagement?
- In today's networked society, harnessing the power of human connections is key. How can we make data conversational and sharable?
- Beyond the desktop visualizations may be effective in engaging hard to reach groups. How can we unlock the hidden potential of tangible data?
- Smartphones and other connected devices bring visualization closer to people than ever before. How can we leverage the ubiquity of data in people's lives?

References

- [Ber67] BERTIN J.: *Sémiologie graphique: les diagrammes, les réseaux, les cartes*. Gauthier-Villars, Paris, 1967. [2](#)
- [Boh14] BOHMAN S.: Information technology in eParticipation research: a word frequency analysis. In *Electronic Participation*, Tambouris E., Macintosh A., Bannister F., (Eds.), no. 8654 in Lecture Notes in Computer Science. Springer Berlin Heidelberg, Jan. 2014, pp. 78–89. [1](#)
- [Cai15] CAIRO A.: Graphics lies, misleading visuals. In *New Challenges for Data Design*, Bihanic D., (Ed.). Springer London, 2015, pp. 103–116. [3](#)
- [Cha08] CHADWICK A.: Web 2.0: new challenges for the study of e-democracy in an era of informational exuberance. *I/S: A Journal of Law and Policy for the Information Society* 5 (2008), 9. [1](#)
- [CM12] COLEMAN S., MOSS G.: Under construction: the field of online deliberation research. *Journal of Information Technology & Politics* 9, 1 (2012), 1–15. [1](#)
- [CMS99] CARD S. K., MACKINLAY J. D., SHNEIDERMAN B.: *Readings in information visualization: using vision to think*. Morgan Kaufmann Series in Interactive Technologies, 99-3236956-X. Kaufmann, San Francisco, 1999. [2](#)
- [FD12] FOX S., DUGGAN M.: *Mobile health 2012*. Tech. rep., Pew Research Center, 2012. <http://www.pewinternet.org/2012/11/08/mobile-health-2012/>. [3](#)
- [Fri08a] FRIENDLY M.: A brief history of data visualization. In *Handbook of Data Visualization*, Springer Handbooks of Computational Statistics. Springer Berlin Heidelberg, Jan. 2008, pp. 15–56. [2](#)
- [Fri08b] FRIENDLY M.: The golden age of statistical graphics. *Statistical Science* 23, 4 (Nov. 2008), 502–535. [2](#)
- [HCL05] HEER J., CARD S. K., LANDAY J. A.: Prefuse: a toolkit for interactive information visualization. In *Proc. of SIGCHI* (New York, NY, USA, 2005), CHI '05, ACM, pp. 421–430. [2](#)
- [Jan09] JANSEN W.: Neurath, Arntz and ISOTYPE: The Legacy in Art, Design and Statistics. *Journal of Design History* 22, 3 (Sept. 2009), 227–242. [2](#)
- [KA13] KATZ J., ANDREWS W.: How y'all, youse and you guys talk, Dec. 2013. <http://www.nytimes.com/interactive/2013/12/20/sunday-review/dialect-quiz-map.html>. [2](#)
- [KM13] KOSARA R., MACKINLAY J.: Storytelling: the next step for visualization. *Computer* 46, 5 (May 2013), 44–50. [2](#)
- [Kru13] KRUM R.: *Cool infographics: effective communication with data visualization and design*. John Wiley & Sons, 2013. [3](#)
- [MCS09] MACINTOSH A., COLEMAN S., SCHNEEBERGER A.: eParticipation: the research gaps. In *Electronic Participation*, Macintosh A., Tambouris E., (Eds.), vol. 5694 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Jan. 2009, pp. 1–11. [1](#)
- [Moe08] MOERE A. V.: Beyond the tyranny of the pixel: exploring the physicality of information visualization. In *Information Visualisation, 2008. IV'08. 12th International Conference* (2008), pp. 469–474. [3](#)
- [MS14] MAYR E., SCHREDER G.: Isotype visualizations. A chance for participation & civic education. *JeDEM - eJournal of eDemocracy and Open Government* 6, 2 (Feb. 2014), 136–150. [2](#)
- [New14] NEW YORK TIMES: The New York Times's most visited content of 2013, Jan. 2014. <http://www.nytimes.com/the-new-york-times-most-visited-content-of-2013/>. [2](#)
- [Ste12] STEFANER M.: Data cuisine | Exploring food as a form of data expression, Sept. 2012. <http://www.data-cuisine.net.3>
- [Tuf83] TUFTE E. R.: *The visual display of quantitative information*. Graphic Press, Cheshire, Conn., 1983. [2](#)
- [Tuk77] TUKEY J. W.: *Exploratory data analysis*. Addison-Wesley, Reading, Mass., 1977. [2](#)
- [WK15] WOLF G., KELLY K.: Quantified self – self knowledge through numbers, Feb. 2015. <http://quantifiedself.com>. [3](#)

Exploring Time Relations in Semantic Graphs

Ralph Wozelka^{1,*}, Mark Kröll^{1,*} and Vedran Sabol^{1,2}

¹Know-Center GmbH, Inffeldgasse 13, Graz, Austria

²Graz University of Technology, Inffeldgasse 13, Graz, Austria

{rwozelka|mkröll|vsabol}@know-center.at

ABSTRACT

The analysis of temporal relationships in large amounts of graph data has gained significance in recent years. Information providers such as journalists seek to bring order into their daily work when dealing with temporally distributed events and the network of entities, such as persons, organisations or locations, which are related to these events. In this paper we introduce a time-oriented graph visualisation approach which maps temporal information to visual properties such as size, transparency and position and, combined with advanced graph navigation features, facilitates the identification and exploration of temporal relationships. To evaluate our visualisation, we compiled a dataset of ~120.000 news articles from international press agencies including Reuters, CNN, Spiegel and Aljazeera. Results from an early pilot study show the potentials of our visualisation approach and its usefulness for analysing temporal relationships in large data sets.

Categories and Subject Descriptors (according to ACM CCS): H.3.3 [Information Search and Retrieval]: Information filtering; H.5.2 [User Interfaces]: Graphical user interfaces (GUI).

1. Motivation

The processing and analysis of information is becoming more and more complex partly due to the vast amount of available information, but also due to our capabilities of automatically explicating relations within the data. This relationship analysis has already found its way into some profession's everyday work including journalists or historians which seek to get a chronological overview of a certain topic. To adequately represent relational data, node-link-based graph representation is widely used. Its simplicity and established formalism facilitate the analysis of relationships between data nodes. Adding a temporal component to graph visualisation, and with it an additional source of complexity, is not easy to achieve since complex graph representations already have a tendency towards clutter due to link crossing and overdraw.

In this paper, we extend the Semantic Blossom graph [RWV*14], an approach supporting focused and context-sensitive exploration of complex graphs along semantic facets, by mapping temporal information to visual properties such as (i) node size, (ii) transparency and (iii) position. Interactive combination of these properties facilitates the identification and thus the exploitation of temporal relationships in the data.

2. Related Work

Network data such as semantic knowledge graphs are commonly visualised using node link diagrams and graph visualisation methods (cf. [Cui07]). The representation of complex graph structures exhibits a tendency towards clutter due to link crossing and overdraw; adding a temporal component to it is thus not an easy to achieve goal. A commonly taken approach of addressing graph evolution is by showing changes through an animated sequence (cf. [BPF*14], [Raf13]). While animation may appear a natural way of conveying change, it suffers from the phenomenon of change blindness and from the human inability to remember previous states. Another approach involves visualising the graph at different time points through multiple snapshots or temporal slices [CKN*03]. While this group of methods enables the viewer to compare multiple graph states within a single static visualisation, and to view development of specific graph parts over time [Dwa05], they add a large amount of complexity to the representation as the graph is drawn multiple times. Additionally, slices are often shown in 3D [FHN*07] which complicates navigation and may lead to occlusion problems. Some approaches combine temporal slicing with animation [EHK*05]. A related approach emphasises the temporal dimension over the graph structure by positioning the nodes along a timeline [BPO*06]. As the connectivity of the graph does not contribute to the layout, such a representation is often cluttered by links connecting nodes placed far from each other, making such a layout suitable only for smaller graphs. A very commonly taken approach is to code the temporal information using visual features of nodes and links, such as colour or transparency [ATS*11]. Nevertheless, the advantage of representing the history of the whole graph at once within a single, stable representation may be outweighed by the complexity of the representation for large graphs. This is due to the fact that all nodes and links, i.e. those within and those outside of the user's time interval of interest, are always displayed.

* both authors contributed equally to this work

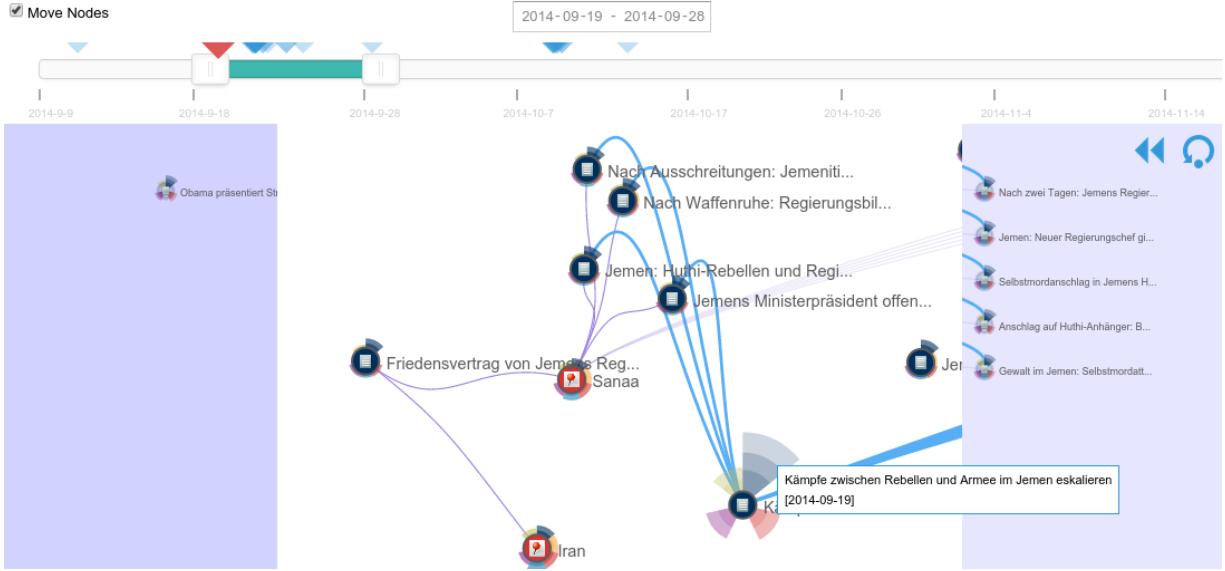


Figure 1 gives an overview of the visualisation interface. The main part of the interface consists of three views – a past view (left), a main view (middle) and a future view (right). The main view contains only nodes whose time stamp fall into the time span selected with the time slider at the top; the selection corresponds to the date information – in this case 2014-09-19 – 2014-09-28. Nodes before the selection are displayed in the past view; nodes after the time span in the future view. All displayed nodes are assigned a corresponding triangle icon on the time slider representing their time stamp, with selected node’s time shown in red. A “mouse over” highlights a node, provides more comprehensive information in a tool tip, in this case the article’s full title as well as the publication date, and shows the node’s semantic blossom which is used for focused and context-sensitive exploration of the graph.

Our work combines the advantages of coding temporal information using visual features, in particular, size and transparency, with positioning of nodes which are outside of the user selected time interval at the edges of the visualisation. In this way the user can focus on the part of the graph within the selected time interval which is shown as a classical node-link representation with full exploration capabilities. At the same time the context of the full graph and the temporal information remain preserved.

3. Mapping Temporal Information

The used data set contains ~120.000 news articles from international press agencies including Reuters, CNN, Spiegel and Aljazeera. The articles were in English and German with matching on syntactic but not on semantic level. The articles were crawled from March, 2014 to January, 2015. Each article’s time stamp represents its publication date. These articles are then processed using NLP methods to extract named entities (such as persons, locations and organisations) and stored into an index. Associations between extracted named entities are computed based on their co-occurrence and stored into an association index (cf. [Ker12]), which is implemented as a double inverted index (concept – documents, document – concepts). Queries to the association index form the basis for visual exploration of relationships.

We base our visualisation on the Semantic Blossom graph [RWV*14]), a graph visualisation approach which

facilitates the exploration of large densely connected graphs in a smart and efficient way by initially showing not the entire graph but only a subset of relevant nodes. Users can expand nodes along semantic facets to explore the graph. We extend this approach by a temporal component which maps a time information attached to nodes onto visual properties such as (i) node size, (ii) transparency and (iii) position. We use multiple visual properties to emphasise the difference between nodes in and outside the selected time interval. Using only a single visual property, e.g. size, would either make the symbol too small to be readable or would not distinguish the two groups adequately.

Figure 1 gives an overview of our interface to visualise the temporal information within the data. Relevant time spans can be selected by handles on the time slider at the top (the minimal time span is 48 hours). This temporal selection triggers the assignment of nodes to three views – a Past view, a Main (Present) view and a Future view. Nodes with a time stamp in the selected time span are shown in the Main view using a node-link representation. All displayed nodes are assigned a triangular icon above the time slider according to their time stamp; a selected node is marked by a red triangle. A “mouse over” highlights a node and provides more comprehensive information; in this case the article’s full title as well as the publication date. The highlighted node illustrates the concept of the semantic blossom that shows which and how many relations exist to distinct types of nodes using a “blossom” representation. The colour (and position) of

each petal defines the node type to expand, its size indicates the number of connected nodes of that particular type. Context-sensitive expansion, i.e. only of those nodes which are connected to specified nodes already shown in the visualisation, is possible via simple interactions with petals and nodes.

3.1 Node Positioning

Nodes outside of the selected time span are temporally sorted and displayed in their respective views. This means that nodes closest to the temporal selection are positioned at the top of their views, whereby their size as well as their opacity also increase (see Figure 2).



Figure 2 shows the temporal ordering of nodes in the Future view. The closer to the Main view in terms of time, the larger and the more opaque they are. Temporally more distant nodes are smaller and more transparent. Edges are preserved between nodes in different views; to prevent clutter, they are indicated using highly transparent links.

By ticking off the “Move Nodes” check box, all nodes are positioned in the Main view. In this case the temporal information is only conveyed by node size and node transparency as illustrated in Figure 3.

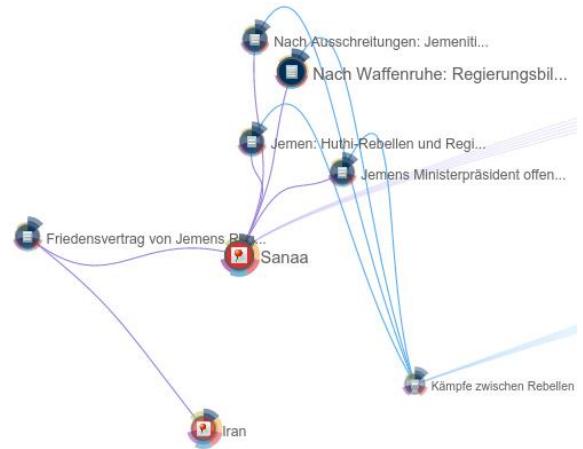


Figure 3 shows the visualisation in a “Move Nodes” ticked off mode. All displayed nodes moved to the main view; temporal information is conveyed only by node size and node transparency. In this case, large and opaque nodes exhibit time stamps within the temporal selection. The smaller and more transparent a node is, the further it is from the selected time interval.

It should be noted that temporal information is different for article nodes and facet nodes. Article nodes have a single time stamp. Facet nodes can either represent named entities, e.g. persons, locations or organisations, which are contained in an article or metadata such as source information. Since facet nodes are not unique, i.e. a person can occur in more than one news article, they can have more than one time stamp as illustrated in Figure 4. A “mouse over” in the graph view reveals time stamps in the timeline of all other displayed articles the person “Mohammed Basindawa” occurs in. The time stamps are furthermore highlighted at the time slider via red triangles facilitating to identify temporal relationships amongst the nodes.

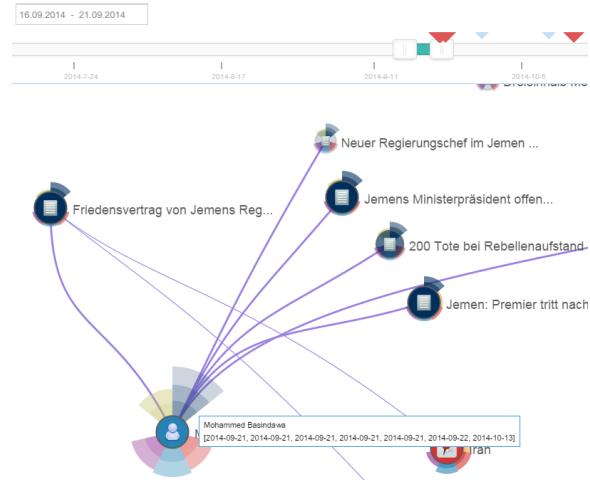


Figure 4 shows the “mouse over” information of a person facet node (“Mohammed Basindawa”) as well as the time stamps of all other displayed articles which mention this person. These time stamps are also marked via red triangles on the time slider. For a possible positioning in the views the average of the time stamps is taken.

4. Pilot Study

4.1 Task Description

We conducted an early pilot study with two participants in form of a thinking aloud test to collect initial feedback on usability problems and on whether one of the modes would be preferred by the users. The pilot study also serves as a preparation for a subsequent full-scale user study, which will compare the two modes of temporal representation – with and without moving nodes – in detail. The two participants had experience in gathering information; both search daily to once a week for information on people and organisations on the web using a standard search engine.

The evaluation task required the participants to look into the conflict going on in the past months in the Republic of Yemen. Starting point was the search query “Yemen” resulting in the most relevant documents being displayed in our visualisation interface. The two participants were steered through a prepared scenario by posing questions such as “Find out the name of the related Yemen politi-

cian” or “When was the cease-fire between rebels and government reported?”. At the end, a big picture of the Yemen conflict should have been formed taking into account chronological ordering of the events: “fighting”, “peace treaty”, “elections” and “prime minister resigns”.

4.2 Feedback and Observations

In the following we report selective findings from the thinking aloud test as well as the filled out questionnaires at the end of the pilot study:

- Having the distribution of timestamps visualised (triangles) above the time slider was helpful.
- Highlighting the triangle icons above the slider on node “mouse over” was perceived as a very useful inspection helping find timestamps and establish a temporal context to other nodes.
- The “Move Nodes” representation was easier to read due to a better focus on the selected time interval and less clutter, which is caused by many out-of-context nodes and links in the other mode.
- The “Move Nodes” mode made it easy to overlook new nodes as they were added and moved out of the way.
- If nodes move, context is lost more easily. The movement itself needs to be slow and smooth to avoid confusion.
- The currently missing synchronised highlighting over all views in the “Move Nodes” mode would be helpful.
- Visualisation design is generally suitable for discovering most important relationships in the data.
- The participants agreed that the visualisation, in both modes, supports discovery of temporal information, for instance, when, or how far apart events happened.

To summarise: moving nodes helps avoid clutter, which is welcome, but new nodes added through expansion tend to be overlooked easily, which will be addressed in a future version of the tool. It is also an argument for providing both modes to the user.

5. Conclusions

In this work we presented an approach for visualising time-oriented data by mapping temporal information to visual properties, in particular the node position, however taking a middle approach where only nodes out of the user-defined temporal focus are moved, while the within-focus nodes remain unaffected. Our visualisation enables and facilitates explorative and explanatory analyses of time-oriented data graph structures, for instance, for information providers to get an overview of a situation based on (i) respective events in the past or (ii) discovered relations between data entities such as persons or organisations.

To gather evidence on the effectiveness of our approach, we plan to conduct a comprehensive user study with up to 30 professional with information analysis background. Future work also includes an improvement with respect to the user interface, for instance, implementing a proper edge rendering between nodes from different views.

Acknowledgement. This work is funded by the FFG KIRAS program (project nr. 840824). Know-Center is funded within the Austrian COMET Program under the auspices of two Austrian Federal Ministries - BMVIT and BMWFW - and State of Styria. COMET program is managed by the FFG.

References

- [ATS*11] AHN J., TAIEB-MAIMON M., SOPAN A., PLAISANT C., SHNEIDERMAN B.: Temporal visualization of social network dynamics: Prototypes for nation of neighbors. In *Proc. of the 4th International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction*, (2011).
- [BPF*14] BACH B., PIETRIGA E., FEKETE J.: GraphDiaries: Animated transitions and temporal navigation for dynamic networks. *IEEE Transactions on Visualization and Computer Graphics*, (2014).
- [BPO*06] BLYTHE J., PATWARDHAN M., OATES T., DES JARDINS M., RHEINGANS P.: Visualization support for fusing relational, spatio-temporal Data: Building Career Histories. In *Proc. of the 9th International Conference on Information Fusion*, (2006).
- [CKN*03] COLLBERG C., KOBOUROV S., NAGRA J., PITTS J., WAMPLER K.: A system for graph-based visualization of the evolution of software. In *Proc. of the ACM Symposium on Software Visualization*, (2003).
- [Cui07] CUI W.: A survey on graph visualization. PhD Qualifying Exam (PQE) Report, Computer Science Department, Hong Kong University of Science and Technology (2007).
- [Dwa05] DWAYER T.: Extending the wilmascope 3D graph visualisation system. In *Proc. of the Asia-Pacific Symposium on Information Visualisation*, (2005).
- [EHK*05] ERTEM C., HARDING P., KOBOUROV S., WAMPLER K., YEE G.: Exploring the computing literature using temporal graph visualization. In *Proc. of the Conference on Visualization and Data Analysis*, (2005).
- [FHN*07] FU X., HONG S., NIKOLOV N., SHEN X., WU Y., XU K.: Visualization and analysis of email networks. In *Proc. of Asia-Pacific Symposium on Visualisation*, (2007).
- [Ker12] KERN R.: A feature association framework for knowledge discovery applications. PhD Thesis. (2012)
- [Raf13] RAFELSBERGER W.: Interactive visualization of evolving force-directed graphs. Lecture Notes in Computer Science, (2013).
- [RWV*14] RAUCH M., WOZELKA R., VEAS E., SABOL V.: Semantic Blossom Graph: A New Approach for Visual Graph Exploration. In *Proc. of the 18th Int. Conf. on Information Visualisation (IV2014)*.

Advanced Visualization Techniques for Laparoscopic Liver Surgery

Dimitrios Felekidis¹, Peter Steneteg¹, and Timo Ropinski²

¹ Scientific Visualization Group , Linköping University, Sweden

² Visual Computing Group, Ulm University, Germany

Abstract

In order to make it easier for the surgeons to locate tumors during a laparoscopic liver surgery, and to form a mental image of the remaining structures, the 3D models of the liver's inner structures are extracted from a pre-operative CT scan and are overlaid onto the live video stream obtained during surgery. In that way the surgeons can virtually look into the liver and locate the tumors (focus objects) and also have a basic understanding of their spatial relation with other critical structures. Within this paper, we present techniques for enhancing the spatial comprehension of the focus objects in relation to their surrounding areas, while they are overlaid onto the live endoscope video stream. To obtain an occlusion-free view while not destroying the context, we place a cone on the position of each focus object facing the camera. The cone creates an intersection surface (cut volume) that cuts the structures, visualizing the depth of the cut and the spatial relation between the focus object and the intersected structures. Furthermore, we combine this technique with several rendering approaches, which have proven to be useful for enhancing depth perception in other scenarios.

1. Introduction

During the past years gradually more surgeries are performed using laparoscopic techniques. During the surgery an endoscope camera transmits high quality video streams to high resolution monitors in the operating room allowing the surgeons to perform the same tasks as in open surgery. However, the endoscope video stream only provides visualization of the surface of the organ without giving information for the critical inner structures. Computer Assisted Surgery (CAS) is the concept of performing laparoscopic surgeries using computer technology. In CAS the first step is to create a 3D model that reproduces in great accuracy the geometry of the patient's organ. This can be done through a number of medical imaging technologies, with Computed Tomography (CT) being the most widely used due to its high accuracy. The 3D models reconstructed from CT are then uploaded into the computer system and rendered on screen.

In CAS surgeons also have the ability to use a navigation system that can register the used surgery instruments in relation to the 3D models in real time. By overlaying the 3D models on the live endoscope image, surgeons have the ability to reference the position of the surgical tools in relation to the 3D models and simultaneously with the patient's organ.

However, the representation of the 3D models have to provide a clear view of the critical areas (tumors and cysts), as well as correct depth perception and spatial information between structures. Within this paper we focus on enhancing the visualization of an augmented reality navigation system providing a more focused view of the areas where tumors are placed and better depth and spatial relation between them and their surrounding structures.

2. Related Work

Overlaying the anatomical structures on to the live endoscope image is not sufficient to ensure a good visual result. In computer graphics and especially in medical visualization one of the most important parts is to let the user perceive the spatial relationship between the objects in a 2D display. There are different techniques that allow users to estimate distances such as shading, contours, shadows, aerial perspective, texture gradients etc.. The depth perception is enhanced using halos that highlight the edges of certain structures [BG07] while depth perception is also influenced by different lighting parameters [TI09]. Lighting and shading are important as far as the visual feedback for space and depth perception and the impact of different luminance

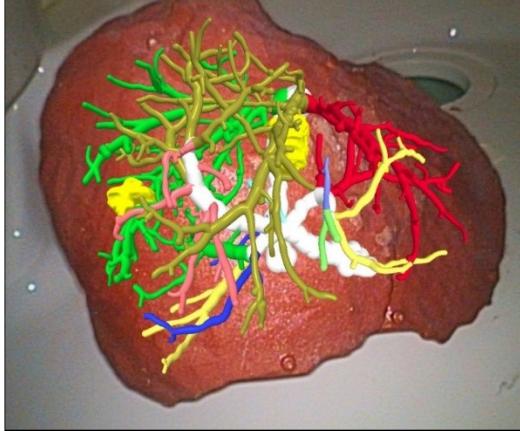


Figure 1: Illustration of the non-enhanced overlay of the 3D models on to the live endoscope image.

patterns are studied. Wagner compared different depth cues and their impact on depth perception [WFG92]. Different techniques that do not depend on transparency which can lead to errors and provide visibility of the focus structures have been developed and are referred to as smart visibility [VG05]. Smart visibility techniques rely on selectively removing structures or cutting into them. These techniques manage to preserve a high quality visual result despite the spatial deformations. Krüger et al. implemented a smart visibility technique by applying ghosting that fades out all the structures that lay in front of the tumors, called Clear View [KSW06]. The visibility to specific structures can be also assured by cutting into a 3D model, a technique that provides also depth perception by visualizing the depth of the cut. Such techniques are referred to as volume cutting. Diepstraten et al. present various approaches to generate cutaways to allow a clear view into a solid object [DWE03]. Two different techniques are presented called cutout and breakaway. The former will remove a big part of the exterior of a geometry to reveal hidden geometries, while the latter will create a cutout in the shape of the hidden object. Rautek et al. used the tumor's silhouette to create the cut's shape providing information for the tumor's shape and the structures that lay in it but not reliable information for the depth of the cut [RBG08]. Using the Maximum Importance Projection the structures are cut by cones to visualize the depth of the cut and also the spatial and depth relation between the structures that lay inside the cones [VKG04].

3. Visualization Enhancements

To improve the non-enhanced overlay of the 3D models on to the live endoscope video stream (see Figure 1), the following visualization techniques were applied with the goal to enhance the depth perception as well as the spatial comprehension of the focus areas.

3.1. Blending and Environmental Depth Cues

In the overlay shown in Figure 1, the 3D models are perceived as the top layer and the endoscope image the bottom layer. It is clearly visible that 3D models seem to float over the liver's surface. In order to give the feeling that the 3D models lay underneath the liver's surface a blending mode has to be applied for both layers. After testing different modes, the screen blend mode managed to produce a result that suits the current case. With screen blend mode the colors of both layers are inverted, then multiplied and then inverted again according to the following equation:

$$f(a, b) = 1 - (1 - a) * (1 - b) \quad (1)$$

where a and b are the two layers. This blend mode does not distort the colors of the two layers, keeps the main color gamut of the scene and also produces a brighter but not too bright image in contrast to the other tested blend modes.

Another downside of the current situation is, that the intensity of the colors of the 3D models as seen in the overlay is the same no matter if they are closer or further from the camera. In computer graphics in order to enhance the depth perception of a scene, techniques that simulate the environmental depth cue (atmospheric perspective) are often used [War12]. That refers to the decrease in contrast of distant objects in the environment. In order to simulate the environmental depth in the laparoscopic scenario, the inverted depth values of the 3D models are added to Equation 1.

3.2. Edge Detection

Due to the applied blending, the inner structures lie now below the liver's surface. Unfortunately, thus also the spatial relation of the 3D structures has become harder to detect due to the blending. By applying an edge detection algorithm and drawing the contours of the structures, the spatial relation can be enhanced. The edge detection is applied before the blending on the rendered 3D models. As the scene is not that complex most of the edge detection algorithms will produce quite a good result [JS09]. A basic average intensity difference edge detection algorithm is applied in a fragment shader. The algorithm traverses all fragments and averages the adjacent intensity values, creating a mask with the averaged RGB colors of all the neighboring pixels. Then, the intensity differences between the neighboring pixels are averaged:

$$I_x = \frac{|I_1 - I_7| + |I_0 - I_8| + |I_3 - I_5| + |I_6 - I_2|}{4} \quad (2)$$

Finally, applying a user defined threshold to the current pixel's final intensity value will determine whether or not that pixel is an edge. The edge detected models are then added in Equation 1:

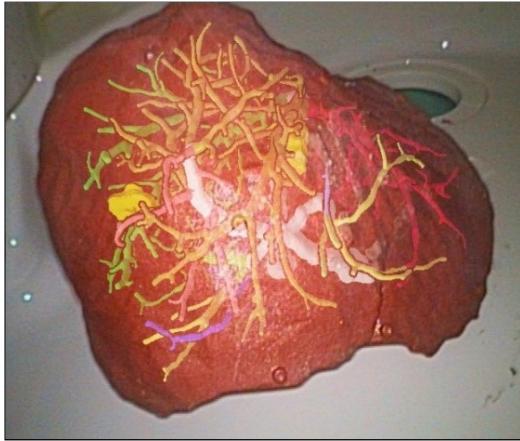


Figure 2: Adjusted screen blend mode using the edge detected models and simulating the environmental depth cue using the inverted depth values.

$$f(a, b) = 1 - (1 - a) * (1 - b_{edge_det} * b_{inv_depth}) \quad (3)$$

The result after applying the described blend mode together with the edge detection can be seen in Figure 2.

3.3. Focus+Context Visualization

During a laparoscopic surgery the surgeons have to focus on the areas where the tumors are located. The location of the tumors, from now on referred to as focus objects, and the proximity to critical surrounding areas such as main blood vessels is extremely important to the surgeons for treatment decisions. The liver contains a very complicated network of vessels that most of the times occlude the focus objects. In order to enhance the depth and spatial perception of the focus objects with their surrounding structures, a cut volume can be created using a cone placed on the positions of each focus object. When a cone is intersected in different spots, the depth comparison between the intersected points is easier due to its angular shape. This technique is referred as breakaways [KTP10]. The cones are placed in the center of each tumor which can be calculated by taking the average of its minimum and maximum x , y and z coordinates. The size/opening for each cut volume is user defined although the smallest possible size should at least contain the whole tumor. To make sure that the cut volumes contain all the structures that lay in them each cone has to be extended. Knowing that the 3D models cannot extend arbitrarily due to the designated shape of the liver, it is sufficient to scale the cones until they almost reach the near plane. However, when the cones are close to one another, they will inevitably overlap and will block the view to the cut volumes. In order to force the overlapping faces not to be rendered when using

OpenGL, a depth test of greater(GL_GREATER) should be used during the rendering phase of the cones, meaning that structures that are further from the camera are rendered instead of the default process of rendering whatever is closer to the camera. However, there are cases where the depth test of greater will not have any effect, for example when only one cut volume is visualized or if the cut volumes are too far to overlap and thus will block the view to the rest of the scene. In order to size the cut volumes accordingly, the surface of the 3D liver model is used to set their borders. A depth test of greater should be performed when rendering the cut volumes and the liver's surface which will lead to only rendering the sized cut volumes. Thus, the cut volumes are adjusted properly and block as less as possible from the structures that do not lay in them as seen in Figure 3.

4. Results

The final step would be to visualize the cut volumes and the structures that lay in them according to Section 3.3 and the rest of the scene according to Section 3.1 and Section 3.2. Moreover, the cut volumes' color should be as close as possible to the actual liver's color in order to create a more even result. However, the live endoscope image has a lot of different gradients due to the liver's structure, the specular light from the endoscope's light source and of course areas that are not lit properly. Testing a lot of different approaches to color the cut volumes has led to the conclusion that it is sufficient to sample the color of any pixel that approaches the main color gamut of the liver. The final composition can change according to our preferences. We can visualize all the cut volumes at the same view, isolate a cut volume at a time, change the size of all or specific cut volumes and finally add some transparency on them to be able to track the structures that lay behind them. Figure 4 shows a composition when all

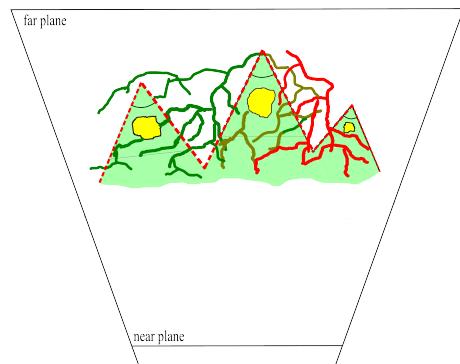


Figure 3: Simulation of the properly sized cut volumes marked in red strip lines. The green contour shows the liver's surface that was used to size the right most and left most faces. The two middle overlapping faces were removed when rendering only the cones with a depth test of greater.

the cut volumes are visualized with different sizes and a low transparency level.

Additionally, the developed implementation can be applied to the 3D liver's surface instead of the live endoscope image. In Figure 5 one cut volume is visualized with a semi-transparent surface.

5. Conclusions

Modern systems are capable of overlaying the pre-operatively extracted 3D models of the patient's liver inner structures, onto the live endoscope video stream during a laparoscopic liver surgery. However, by simply overlaying the 3D models on the live endoscope's video stream is not sufficient to provide a highly accurate spatial and depth perception. Thus, a cone is placed on the position of each tumor facing the camera. Due to the cone's angular shape it is easier to rate the depth of the cut and the distances between the intersected structures. The 3D models that do not lay inside any of the cut volumes are passed through an edge-detection algorithm and blended with the live endoscope image. In that way they still preserve and depict their spatial relation with the rest of the structures. Future considerations in order to achieve a better result would be to add shadows and/or distance lines in the form of rings across each cone's surface.

Acknowledgements

The presented concepts have been developed and evaluated in the Inviwo framework (www.inviwo.org), and have been interfaced with the CAScination navigation system and an augmented reality framework developed at the ARTORG Center for Computer-Aided Surgery.

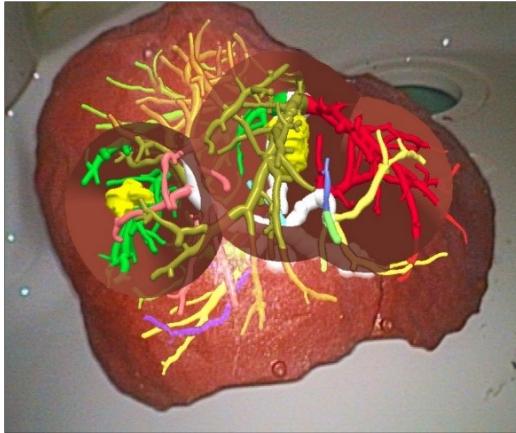


Figure 4: Visualization of all the cut volumes set with different sizes to give a better spatial understanding of the structures. The cut volumes have been set semi-transparent to track the structures that lay behind them.

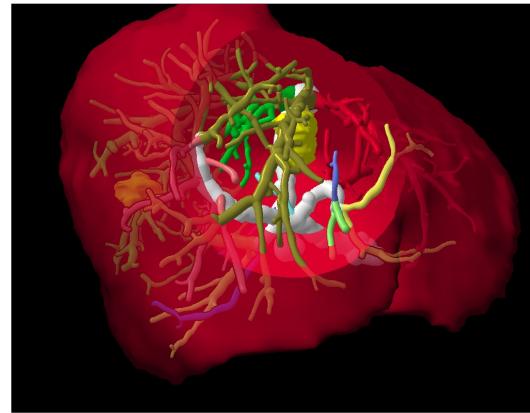


Figure 5: The proposed technique using the 3D model of the liver instead of the endoscope image, isolating one cut volume. Transparency is used for both the cut volume and the liver's surface.

References

- [BG07] BRUCKNER S., GROLLER M.: Enhancing depth-perception with flexible volumetric halos. *Visualization and Computer Graphics, IEEE Transactions on* 13, 6 (2007), 1344–1351. [1](#)
- [DWE03] DIEPSTRATEN J., WEISKOPF D., ERTL T.: Interactive cutaway illustrations. In *Computer Graphics Forum* (2003), vol. 22, Wiley Online Library, pp. 523–532. [2](#)
- [JS09] JUNEJA M., SANDHU P. S.: Performance evaluation of edge detection techniques for images in spatial domain. *methodology I*, 5 (2009), 614–621. [2](#)
- [KSW06] KRUGER J., SCHNEIDER J., WESTERMANN R.: Clearview: An interactive context preserving hotspot visualization technique. *Visualization and Computer Graphics, IEEE Transactions on* 12, 5 (2006), 941–948. [2](#)
- [KTP10] KUBISCH C., TIETJEN C., PREIM B.: Gpu-based smart visibility techniques for tumor surgery planning. *International journal of computer assisted radiology and surgery* 5, 6 (2010), 667–678. [3](#)
- [RBG08] RAUTEK P., BRUCKNER S., GRÖLLER M. E.: Interaction-dependent semantics for illustrative volume rendering. In *Computer Graphics Forum* (2008), vol. 27, Wiley Online Library, pp. 847–854. [2](#)
- [TI09] TAI N., INANICI M.: Depth perception as a function of lighting, time, and spatiality. In *Illuminating Engineering Society (IES) 2009 Conference* (2009). [1](#)
- [VG05] VIOLA I., GRÖLLER E.: Smart visibility in visualization. In *Computational Aesthetics* (2005), pp. 209–216. [2](#)
- [VKG04] VIOLA I., KANITSAR A., GROLLER M. E.: Importance-driven volume rendering. In *Proceedings of the conference on Visualization'04* (2004), IEEE Computer Society, pp. 139–146. [2](#)
- [War12] WARE C.: *Information visualization: perception for design*. Elsevier, 2012. [2](#)
- [WFG92] WANGER L. C., FERWERDA J. A., GREENBERG D. P.: Perceiving spatial relationships in computer-generated images. *IEEE Computer Graphics and Applications* 12, 3 (1992), 44–51. [2](#)

Teaching OpenGL and Computer Graphics with Programmable Shaders

J. Nysjö¹ and A. Hast¹

¹Centre for Image Analysis, Dept. of Information Technology, Uppsala University, Sweden

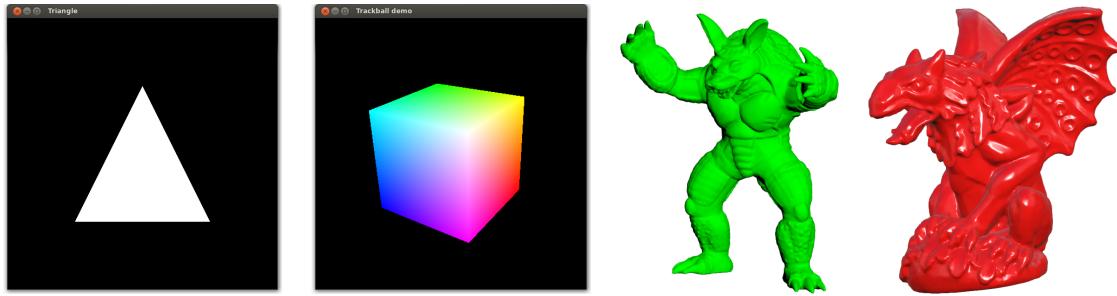


Figure 1: Our course assignments are all implemented with modern shader-based OpenGL and range from manipulating the vertices and fragments of a single triangle to implementing a 3D model viewer with per-fragment Blinn-Phong shading.

Abstract

This paper presents our approach and experiences of transferring an introductory computer graphics course from the fixed-function OpenGL pipeline to modern shader-based OpenGL. We provide an overview of the selected course structure and the C++-based programming environment that we use for assignments and projects, and discuss some of the technical and pedagogical challenges, e.g., multiplatform support and shader debugging, that we ran into. Based on course evaluations and the outcome of programming assignments, we conclude that introducing shaders early and skipping the fixed-function pipeline completely is a sound and viable approach. It requires more initial effort from teachers and students because of the added complexity of setting up and using shaders and vertex buffers, but offers a more interactive and powerful programming environment, which we believe helps promoting the creativity of students.

Categories and Subject Descriptors (according to ACM CCS): I.3.0 [Computer Graphics]: General—

1. Introduction

The transition from fixed-function OpenGL to programmable shaders has slowly reached academia and prompted a change in how graphics programming is taught [AS11]. One of the initial challenges in teaching an introductory computer graphics course based on modern shader-based OpenGL (which here means OpenGL version 3.x or higher) is to help the students overcome the hurdles of compiling shaders and uploading vertex data to the GPU via buffer objects. Another challenge is to set up a programming environment that supports the learning process and works on

a variety of platforms and GPU configurations [PPGT14]. This paper presents our approach and experiences of adopting modern shader-based OpenGL in the introductory computer graphics course offered at Uppsala University. The course, which is supposed to represent 10 weeks of full-time studies and usually have between 50 and 70 enrolled students, is targeted to Bachelor's and Master's CS or engineering students who have taken basic courses in linear algebra and computer programming. Typically, a few of the students have prior experience of shader programming or the fixed-function OpenGL pipeline, but most are complete novices.

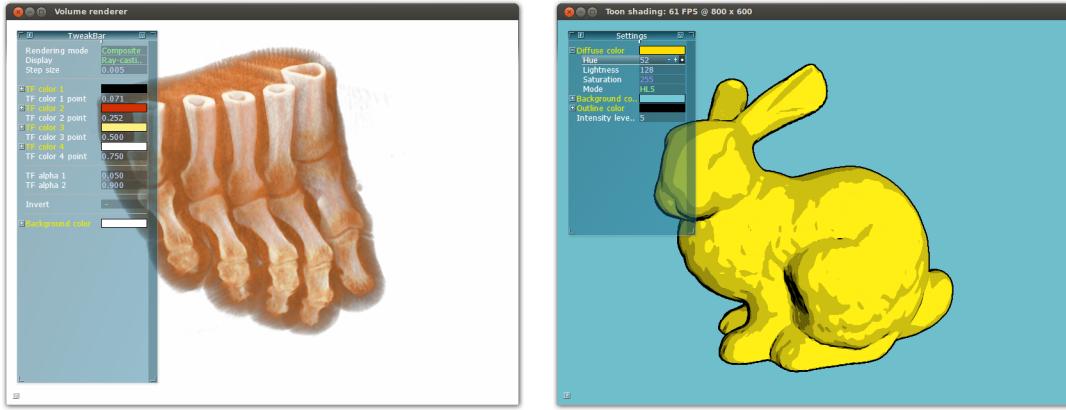


Figure 2: Two of the six available course projects students may choose from. Left: Volume rendering. Right: Toon shading implemented as a post-processing effect. The AntTweakBar GUI library is used for interactive parameter tweaking.

2. Course structure

Whereas shader programming was introduced as an advanced topic towards the end of our former graphics course, we now introduce it already in the second lecture, along with a simplified overview of how the programmable graphics pipeline works. We start by showing how to manipulate vertices and fragments via shaders and how to upload vertex data to the GPU memory via buffer objects, and then move on to cover transformations, viewing, and shading. Subsequently, we introduce more advanced rendering topics such as texture mapping, global illumination, and splines, along with fundamental topics such as rasterization and clipping. The fixed-function OpenGL pipeline is briefly mentioned during the course but not covered in detail.

Our course includes three programming assignments (Fig. 1) and one project. While the assignments specify exactly what the students are supposed to do to assure that they are familiar with the fundamental graphics programming tools and concepts, the project is more loosely specified. The students can choose from six different projects, which cover more advanced rendering techniques such as volume rendering and post-processing effects (Fig. 2). The projects have different degrees of specification so that the students can choose a project according to their own interest but also according to how much freedom they desire and how much specification they need. We also encourage students to suggest projects on their own. In addition to implementing a working solution, each student or project group must submit a 1-2 pages graphical abstract and a short (\sim 1 minute) video demonstrating the solution. The compact format of the presentation enables quick assessment of the quality of the solution, which is appealing from an instructors point of view.

At the end of the course, there is a conventional exam testing the students' theoretical knowledge about the various rendering topics covered in the lectures. The final grade is de-

termined by the grade on the exam and by eventual bonus points obtained in the assignments.

3. Programming environment

The assignments and project are programmed in C++ using the widely supported OpenGL 3.2 core profile and GLSL 1.50. Although we would have preferred to use a compatibility profile, the 3.2 core profile is the only reasonable modern OpenGL profile that is supported under Windows, Linux, and Mac OS X on up to five years old integrated or discrete GPUs. The students can work either on their own computers or on the lab computers, which are equipped with Windows 7, Visual Studio 2013, and Nvidia 6XX series GPUs. We use the FreeGLUT library for creating and managing windows (with GLFW as alternative for Mac OS X users), GLEW for loading OpenGL extensions, and GLM [Cre15] for mathematics. We also use the easy-to-integrate AntTweakBar GUI library [Dec15] to set up a widget for interactive parameter tweaking. Compilation on different platforms is enabled via the CMake [Kit15] build tool, which can generate Visual Studio project files on Windows and Unix makefiles on Linux or OS X.

Compared with the fixed-function OpenGL pipeline, modern shader-based OpenGL has, as noted in [AS11], a steeper learning curve and requires more initial programming to display something on the screen. One of the initial barriers for newcomers to modern OpenGL is to perform the somewhat complex and error-prone steps of loading and compiling shader programs and setting up vertex buffer and vertex array objects. In the first assignment, we provide utility functions for performing these tasks. This allows students to get started quickly and focus on the shader programming, without abstracting away too much of the underlying graphics API. In our experience, wrapping OpenGL calls into a custom framework of C++ classes tend to confuse students

with limited programming and C++ background. Thus, we prefer to use plain OpenGL calls, for which the students easily can find documentation and examples. The only additional utility code we provide for the assignments is a virtual trackball implementation and a simple OBJ file reader.

4. Challenges

Many of our students find it difficult to debug OpenGL applications and shader code. Shader variables can not be easily printed, and sometimes it is not obvious whether the problem lies in the host application code or in the shader code. Commenting out code and rendering vector or scalar variables as colors to the screen are the typical shader debugging techniques that we suggest for the assignments. Adding a keyboard shortcut for reloading shader programs on-the-fly is also highly useful since it allows the students to interactively modify shaders without restarting the OpenGL host application. The `ARB_debug_output` extension can be enabled to facilitate debugging on the host side. Some students have found graphical debuggers like APITrace useful later in the course.

Live coding during the initial lectures is helpful to illustrate how shaders can affect the appearance of the rendering and how the different types of shader variables are used. As a supplement to the course material, we also encourage the students to work through some of the many excellent modern OpenGL programming tutorials (e.g., [dV15]) that are available online. To avoid that the students try to use legacy OpenGL functions in the assignments, we provide a quick-reference listing the most common fixed-function OpenGL commands that are deprecated or removed in the OpenGL 3.x and 4.x core profiles.

The programming environment described in Section 3 can be a bit complex to set up for the first time. CMake is less ideal for the GUI-based workflow on Windows, and some of the students who worked on Mac OS X ran into issues with third-party libraries. A possible future direction of the course could be to move away from desktop OpenGL and C++ to WebGL and JavaScript, so that the students only would require a text editor and a WebGL-enabled browser to develop and run their OpenGL applications. This approach has been successfully adopted in, for example, massive open online courses (MOOC) [Cou15].

5. Conclusion

According to Romeike [Rom08] there are three drivers for creativity in computer science education: 1) the person with his motivation and interest, 2) the IT environment, and 3) the subject of software design itself. Allowing the students to choose a project is promoting the first driver. We strongly believe that using programmable shaders instead of the old fixed-function OpenGL pipeline promotes the second driver. Having control of powerful shaders is simply much more fun

as it allows the student to try out new ideas quickly and easily and implement more sophisticated rendering techniques. The third driver is promoted by the fact that the response is immediate and graphical. The student instantly sees the result on the screen from changing the code and that itself helps the student in the exploration of different parameters and studying the result of code changes.

This is the second year we teach the revisited course, and overall the transition from the old fixed-function OpenGL pipeline to modern shader-based OpenGL has been a positive experience. Based on course evaluations, student feedback, and the outcome of programming assignments, we conclude that introducing shaders early and skipping the fixed-function pipeline completely is a sound and viable approach, particularly since the OpenGL 3.2 core profile is now widely supported. Although the shader-based approach requires more initial effort from both teachers and students, it appears to provide the students a more fundamental understanding of how the graphics pipeline works, smoothens the transition to advanced rendering topics, and pays off later in the form of more spectacular course projects. Key issues to address in the future are improved support for shader debugging and simplification of the programming environment.

6. Acknowledgments

The 3D models are courtesy of the Stanford 3D Scanning Repository and the AIM@SHAPE repository. The volume dataset is courtesy of Philips Research.

References

- [AS11] ANGEL E., SHREINER D.: Teaching a shader-based introduction to computer graphics. *Computer Graphics and Applications, IEEE* 31, 2 (2011), 9–13. 1, 2
- [Cou15] COURSERA: Interactive Computer Graphics with WebGL. <https://www.coursera.org/course/webgl>, 2015. Accessed on April 14, 2015. 3
- [Cre15] CREATION G.-T.: OpenGL Mathematics (GLM). <http://glm.g-truc.net/>, 2015. Accessed on April 14, 2015. 2
- [Dec15] DECAUDIN P.: AntTweakBar. <http://anttweakbar.sourceforge.net/doc/>, 2015. Accessed on April 14, 2015. 2
- [dV15] DE VRIES J.: Learn OpenGL. <http://learnopengl.com/>, 2015. Accessed on April 14, 2015. 3
- [Kit15] KITWARE: CMake. <http://www.cmake.org/>, 2015. Accessed on April 14, 2015. 2
- [PPGT14] PAPAGIANNAKIS G., PAPANIKOLAOU P., GREASSIDOU E., TRAHANIAS P.: glGA: an OpenGL Geometric Application framework for a modern, shader-based computer graphics curriculum. *Eurographics 2014, Education Papers* (2014), 1–8. 1
- [Rom08] ROMEIKE R.: Towards Students' Motivation and Interest: Teaching Tips for Applying Creativity. In *Proceedings of the 8th International Conference on Computing Education Research* (New York, NY, USA, 2008), Koli '08, ACM, pp. 113–114. 3

Introducing Computer Game Technologies in a Mathematical Modelling and Simulation Course

C.E. Peters and J. Hoffman

KTH Royal Institute of Technology, Stockholm, Sweden

Abstract

This paper describes the use of interactive computer graphics and game technologies in a new mathematical modelling and simulation course at KTH Royal Institute of Technology that commenced in January 2015. In order to better engage students in the subject, elements of the course involved real-time physics scenarios using computer game technologies. An important secondary aim in the course was to develop the ability for students to define their own goals in the absence of specific tasks, which culminated in student-led projects. This paper briefly summarises the pedagogical approach, the course structure and presents a sample of student project work.

1. Introduction

The fields of scientific computing and computer graphics share a multitude of models, methods and algorithms, where the main difference often is that while in scientific computing quantitative accuracy is a necessity, in computer graphics, the visual impression and real-time operations are paramount. Models and simulation is becoming increasingly important in science and engineering, used for simulation based experiments, virtual prototyping, forecasts and risk analyses. As computer hardware becomes more powerful, there is a gradual move towards more realistic simulation models. Traditionally, numerical analysis courses are taught using software packages such as Matlab/Octave, with a focus on very simple model problems to illustrate the basic concepts, and with limited capabilities for interactivity and visualisation. While this may be suitable for some groups of students, it may also deter other groups that lose interest in the mathematical abstraction of the models. This is significant at national and international level, where attracting, engaging and retaining students in science, engineering and mathematics has been an issue of growing concern.

A core aim of the course is therefore to exploit numerous relationships between modelling and simulation approaches and computer games technologies (primarily physics simulation and interactive visualisation) in order to stimulate curiosity and provide intuitive knowledge about the underlying numerical methods, algorithms and physics models that are foundational to both entertainment applications and more realistic applications in serious science and engineering.

2. Background

There has been increasing emphasis over the past decade on the importance of engaging students in Science, Technology, Engineering and Mathematics (STEM) subjects. This has been partly due to fears of skills shortages, especially in Europe and the US, and has been reflected by a general decline in retention in computing programmes. For example, in the UK, concerns about abilities of graduates in STEM areas led to the commissioning of Next-Gen, the Livingstone-Hope Skills Review [LH11] recommending major changes to STEM education in secondary and higher education, which is recognised as a serious issue by the UK government [Gov11]. A number of potential solutions have been proposed, including suggestions that degrees should be made ‘more fun’ and offer multi-disciplinary and cross-disciplinary programs [Car06]. With this in mind, pedagogical practitioners have been exploring new ways in which to make STEM subjects more attractive to students. A number of approaches (see [APH^{*}12] and [Kur09] for examples) involve the use of interactive visualisations and game technologies to better engage students in these subjects.

3. Rationale

Game technologies are therefore relevant to pedagogy in a number of ways. For example, interactive experiments featuring mathematical equations and the impact of simulation approaches and results are well-supported by the real-time, advanced graphics technologies underlying modern computer games. Modern engines also provide tools and com-

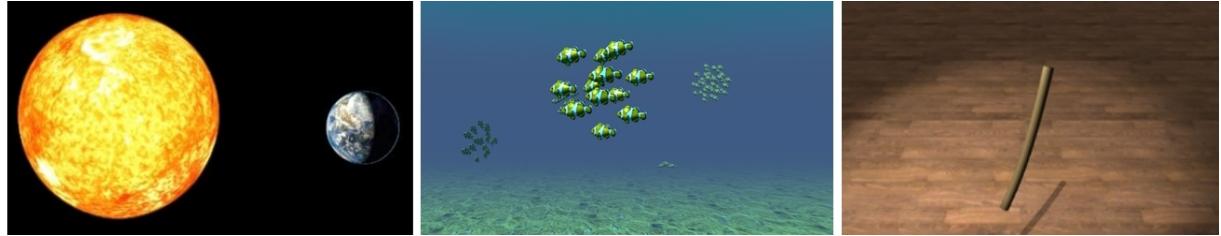


Figure 1: Screenshots taken from three of the lab assignments: (from left to right) solar system, predator-prey and spring mass scenarios, respectively. Lab assignments were designed to become more open ended as they progressed.

ponents for supporting the use of GPUs. Beyond these, one of our primary motivations for introducing computer game technologies in the course relates to stimulating student's intellectual curiosity about nature. Game engines supporting modern AAA bestsellers are exceptionally sophisticated and often aim to simulate and synthesise natural environments that appear realistic to viewers, necessitating the development of underlying physical simulation models. A number of previous works have already investigated the use of the video game technologies to teach physics [Pri08] and numerical methods, for example, to undergraduate mechanical engineering students [CS09], in addition to increasing motivation and engagement in computer science subjects [BPCL08]. This paper presents a preliminary summary of our initial course design and experiences with a cohort of 3rd year computer science students in a 6 week time period.

4. Course Details

We aimed to present the students with a set of engaging scenarios featuring themes that clearly connected to specific computer game technologies, especially game physics. Students could also include simple game mechanics in their scenarios, although the focus of the course was on game technology use and development, rather than game design issues.

4.1. Objectives

There were two main course objectives. The first related to developing students' intuitive understanding of simulation methods with a focus on the following concepts:

- Accuracy: The appropriateness of various models and approximations for different application areas.
- Stability: Issues relating to the numerical stability of various approaches.
- Performance: Implications of various simulation techniques and approximations on achieving real-time interactive performance.
- Generalisability: Application of core methods to many different target domains.

The second objective related to engaging students as active learners in the learning process [APH^{*}12]. Since many

of the students had not engaged in substantial student-led project work before, i.e projects in which they were fully responsible from the specification phase onwards, the course was split into two phases. The first phase consisted of traditional lecture and lab sessions. Initial lab assignments were designed in order to provide a detailed task decomposition for solving problems. As lab assignments progressed, the questions became less detailed and required students to define their own sub-goals by conducting their own task decomposition. This culminated in the second phase of the course, where students engaged in open ended project work.

4.2. Structure

The course was designed around two sequential phases, a first phase that consisted of a traditional set of lectures and labs, and a second phase that consisted of student-led project work facilitated with feedback from the course team.

4.2.1. Phase 1: Lectures and Labs

The first phase consisted of a four week traditional lecture and lab design. Each week included a single lecture on the theories and methods related to modelling and simulation, in addition to a single lecture related to relevant real-time algorithms from graphics and games. Specifically, theory and methods related to the description of particle models, time-stepping methods, ordinary differential equations, implicit methods, partial differential equations and mesh methods, while the practical lectures focussed on the complementary topics of implementing particle systems, steering algorithms, behavioural animation, and a review of game engines and middleware. There were four lab assignments, each covering the following scenario: solar system, predator-prey, spring-mass system and fluid simulation (see Figure 1).

4.2.2. Phase 2: Projects

The second phase of the course, a further two weeks, consisted of a project. Students could complete the projects in groups of up to three people. Unlike the labs, which had a set structure and questions, students could choose their own project topics. Since the issue of choosing a project topic may be non-trivial for many students, they were asked to

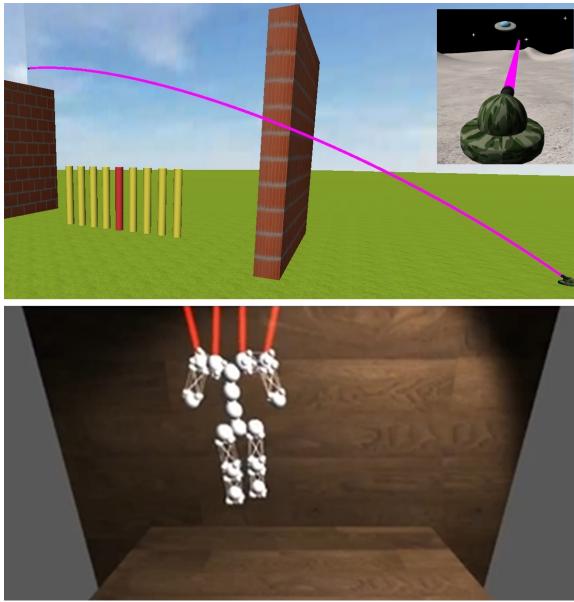


Figure 2: An example of two of the projects from the course, the first (top) involving an interactive cannon ball simulation and (bottom) real-time ragdoll simulation.

submit project specifications that received feedback from the course team. The project phase was supported by two weekly activity and project support sessions. A preliminary and final presentation session also took place where students demonstrated their results and received feedback from the course team and from the rest of the cohort.

5. Results

The course team consisted of the two primary lecturers and four Teaching Assistants (a postdoctoral student and three Master-level students). The 2015 student cohort was composed of twenty 3rd year undergraduate students from the computer science and virtual design programmes in KTH Royal Institute of Technology. A total of eight student groups submitted projects at the end of the course. Figure 2 illustrates two examples of projects from the course. The first project, a cannon ball simulator, introduced game mechanics into a physics scenario in which the player needed to judge wind speed and gravity in order to hit targets in the virtual environment. The second project concerned the implementation of a ragdoll system. Students also took part in a written questionnaire session both at the beginning of the course, to establish their skills, and at the end of the course, to describe their experiences and reflect on approaches for problem solving and project specification. The results generally showed that time management in relation to lab and project submission was a central concern in the course and that most students were motivated not only by achieving spe-

cific grades, but also by creating demonstrations and deliverables that could be used in their future endeavours. It is interesting to note that mathematics and programming issues were not raised as concerns by this cohort.

6. Future Directions

Feedback from the students suggested a number of improvements for subsequent iterations of the course. Firstly, the course is quite short and is intensive in terms of deliverables. Many students were taking other courses at the same time and in some cases this led to time management problems. We also intend to put a further focus on a set of fundamental mathematical models rather than models specific to particular disciplines, for example, focus on wave propagation and dissipation rather than acoustics, electromagnetics, and so on. Another area of future work involves the further separation of the model from the numerical discretisation method, since this abstraction is very powerful.

7. Acknowledgements

We wish to thank the teaching assistants for their help with the course development and delivery: Alejandro Marzinotto, Philip Sköld, Axel Lewenhaupt and Daniel Måansson.

References

- [APH*12] ANDERSON E. F., PETERS C. E., HALLORAN J., EVERY P., SHUTTLEWORTH J., LIAROKAPIS F., LANE R., RICHARDS M.: In at the deep end: An activity-led introduction to first year creative computing. *Computer Graphics Forum* 31, 6 (2012), 1852–1866. [1](#), [2](#)
- [BPCL08] BARNES T., POWELL E., CHAFFIN A., LIPFORD H.: Game2learn: Improving the motivation of cs1 students. In *Proceedings of the 3rd International Conference on Game Development in Computer Science Education* (New York, NY, USA, 2008), GDCSE ’08, ACM, pp. 1–5. [2](#)
- [Car06] CARTER L.: Why students with an apparent aptitude for computer science don’t choose to major in computer science. In *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education* (New York, NY, USA, 2006), SIGCSE ’06, ACM, pp. 27–31. [1](#)
- [CS09] COLLERA B., SCOTT M.: Effectiveness of using a video game to teach a course in mechanical engineering. *Computers and Education* 53, 3 (2009), 900 – 912. [2](#)
- [Gov11] Department for Culture, Media and Sport, UK Government, Government’s response to Next Gen. Transforming the UK into the world’s leading talent hub for the video games and visual effects industries, November 2011. [1](#)
- [Kur09] KURKOVSKY S.: Engaging students through mobile game development. In *Proceedings of the 40th ACM Technical Symposium on Computer Science Education* (New York, NY, USA, 2009), SIGCSE ’09, ACM, pp. 44–48. [1](#)
- [LH11] LIVINGSTON I., HOPE A.: Next-gen report. *Nesta* (February 2011). [1](#)
- [Pri08] PRICE C. B.: The usability of a commercial game physics engine to develop physics educational materials: An investigation. *Simulation and Gaming* 39, 3 (2008), 319–337. [2](#)

A Case Study in Expo-Based Learning Applied to Information Visualization

M. Romero

KTH Royal Institute of Technology, Stockholm, Sweden

Abstract

We present preliminary results of the effect of Expo-Based Learning (EBL) applied to a course on information visualization. We define EBL as project-based learning (PBL) augmented with constructively-aligned large public demos [RTP14]. In this paper, we analyze the results of challenging and grading enrolled students to compete and present their projects publicly at an open student competition organized by a second university. We surveyed the students at the end of the course before the competition started and the end of the competition. We present the result of the impact of the student competition as it relates to the intended learning outcomes from the perspective of the students.

Categories and Subject Descriptors (according to ACM CCS): K.3.2 [K.3 COMPUTERS AND EDUCATION]: Computer and Information Science Education—Computer science education

1. Introduction

Previously, we introduced the concept of Expo-Based Learning as project-based learning constructively-aligned with large public presentations. Constructive alignment refers to assigning a grade for a task that is focused on achieving the intended learning outcomes (ILOs) [RTP14]. In the context of our original presentation of EBL, the students were tasked with presenting in front of two very large audiences at two public events. The course where the students presented was advanced graphics interaction and the events where they presented were a high school student conference with 5,000 attendees and a gaming conference with 30,000 attendance. The students presented for a total of 50 hours at the two events, 10 hours at the first and 40 hours at the second, the 40 hours spread over four days. The technical and communication challenges of these events demonstrated extremely valuable opportunities forwarding deep learning as evidenced by the surveys and individual interviews of the students. To expand on the methodology of EBL, we present the case study in this paper.

We introduce constructively-aligned project competition at an open student competition organized by the Linköping University's C-Visualization Center in Sweden. The event, called the C-Awards [<http://www.cawards.se/>], is a competition open to students from around Sweden. The competition awards 10,000 SEK (1,158 USD) to the winner

of each of six categories. The students in this study sent six projects to compete in five categories.

The information visualization course offers 6.0 credits. During the course, the students complete three individual projects, each worth 20%, and one group project worth 30%. The students also received 10% for weekly readings. This paper focuses on the group project that competed at the C-Awards.

2. Description of the Student Groups, Project Execution, and Grading

The course had eighteen master's students in computer science or human-computer interaction. Four students were female. The students grouped into six teams of three or four people. The grouping criteria was to balance skills across groups and align interests within groups as much as possible.

The content of the project was free for the students to decide. The students had one week to propose and two weeks to implement the projects. They demonstrated the final projects in class, where they received a partial grade. After the presentations, the students had three weeks to improve their visualizations and write a two-page document and record a five-minute video for the competition. At the competition, they were tasked with presenting and demoing their interac-



Figure 1: The students of information visualization during their final project demonstration in the Visualization Studio VIC
<https://www.facebook.com/VisualizationStudio>.

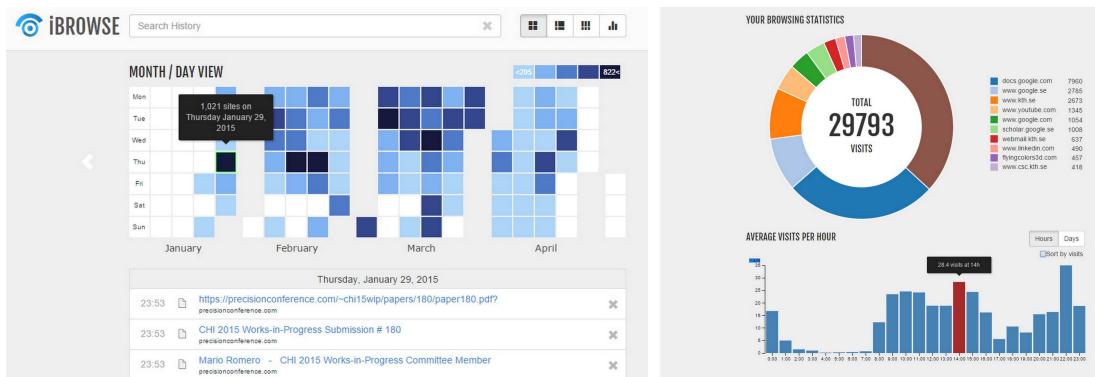


Figure 2: iBrowse is a visualization of internet browsing history. See the video figure <https://vimeo.com/90013649>. Visit the webpage <http://goo.gl/J2kKG4>.

tive visualizations in front of an audience of approximately 400 people.

The projects were graded as follows: 1. Proposal 10%; 2. Mid-term presentation 10%; 3. Final demo 20%; 4. Critiquing other projects 10%; 5. URL with the running code and a description of project 10%; 6. Two-page document and the video submission to the C-Awards 30%; and 7. Two-page report documenting their learning outcomes 10%.

The grading criteria consisted on the quality of the answers to the following questions: 1. Who is the user? 2. What are the tasks? 3. What is the data? 4. What are the data of transformations? 5. What are the visual mappings? 6. What are the visual structures? 7. What are the view transformations? 8. What are the views? 9. How does the demo support the tasks? 10. How can it be improved? The premise of EBL is that flaws in the answer to these questions become evident by presenting the projects to larger and more diverse audiences.

This is a project-based learning course and it is also a studio-based learning. The course took place at the Visual-

ization Studio VIC at KTH Royal Institute of Technology, Stockholm, Sweden. Figure 1 shows the students of information visualization during their final project demonstration in the studio. VIC performs three functions: research, education, and outreach. The research centers on visualization-supported collaborative work and foundational interactive and graphics technologies. It is also a showcase and classroom environment where we teach and demo our existing projects. Finally it is an outreach environment where we invite industry partners and primary and secondary school students to interact with the technology, science, and students at KTH. VIC provides a number of cutting-edge technologies and the expertise to readily use the infrastructure. Some of the technologies VIC houses are an ultra-high definition 4K stereoscopic projection four-meter wall with traditional keyboard and mouse interfaces augmented with gesture-, voice-, touch-, and phone-based interaction infrastructures. VIC houses as well a number of Oculus Rifts, cinema-quality audio, high-definition video conferencing with eye contract, eye tracking, GPU-based computing clusters, diverse interaction and sensor systems, haptic devices, and 3-D printers.

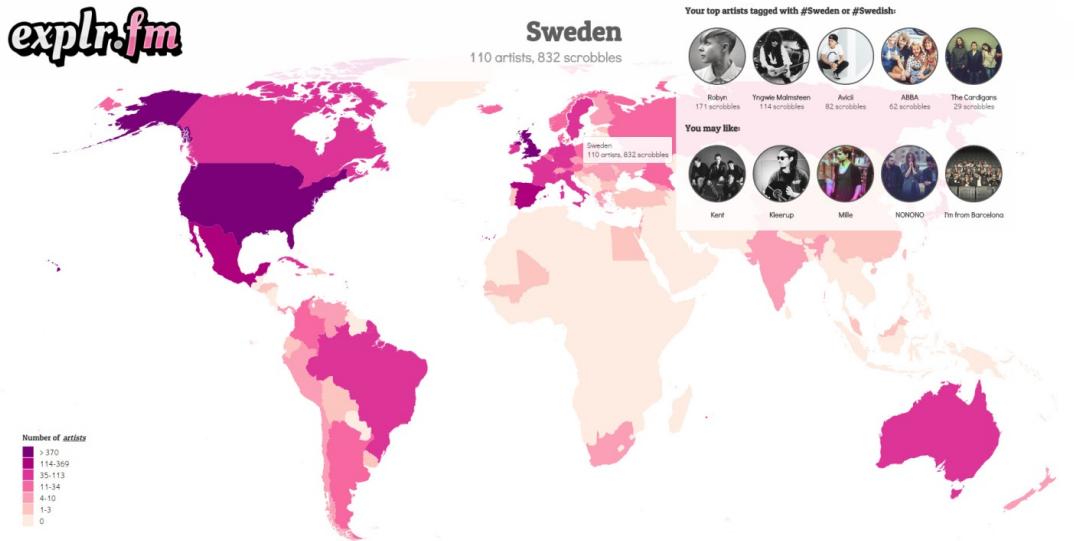


Figure 3: *explr.fm* is a visualization of the national provenance of the artists on a user's LastFM playlist. See the video figure <https://vimeo.com/90011479>. Visit the webpage <http://explr.fm/>.

VIC also has a full-time staff that supports the coordination and engineering of the projects.

3. Two Sample Projects

The course had six group projects. In the interest of space, we highlight two of the six projects here.

Figure 2 shows *iBrowse*, a visualization of internet browsing history [<http://goo.gl/J2kKG4>]. It was created by master's students Wouter Jansen, Ivo van Bon, and Henri Louis Schröter. It shows a calendar view where the color saturation of each cell represents a daily aggregate of the number of site visits. It also provides pie charts and histograms visualizing the most visited web sites and the days and hours were most of the browsing occurs.

Figure 3 shows *explr.fm*, a visualization of the national provenance of the artists on a user's LastFM playlist. It was created by master's students Anna Movin, Daniel Molin, Moa Bergmark, Tommy Feldt. The application provides an overview through a choropleth map and a number of targeted details on demand, including a list of suggestions based on the existing playlist [<http://explr.fm/>]. The project *explr.fm* was the only winner of the six projects in the category people's choice awards.

4. Experimental Design

The goal of the experiment is to measure the impact on intended learning outcomes (ILOs) of the students in information visualization participating at the open student competition, the C-awards. The instrument to measure the impact

is a pre- and post-survey of the perceived learning outcomes by the students. We surveyed the students at the point where they finished presenting the last demo as part of the regular class and then we surveyed the students after their participation at the awards.

We deployed a voluntary online survey. We achieved 100% response rate from the students for both the pre-and the post-surveys. The surveys included Likert scale questions with open-ended fields for expanding on the answer and open-ended questions as well. In the results, we present statistical analysis of the Likert scale questions and we highlight the results of a focused qualitative analysis of the open-ended questions.

5. Results

We first determined the number of hours on task for all the participating students in the course. From the results of the survey, we establish that each student spent an average of 35 hours on the tasks related to the project from its conception to the final presentation in front of the class. Then each student spent, on average, an additional 25 hours between the final course presentation and participating at the C-awards, including the 6 hours at the awards. Participation at the C-awards granted 20% of the grade, yet the students devoted 40% of their time to this part of the project. When asked why this is the case, the most common response was that the competition, the public recognition of winning, the monetary reward, and the risk of humiliating themselves with poor projects motivated them to push themselves harder.

Figure 4 shows the mean and standard error bars from

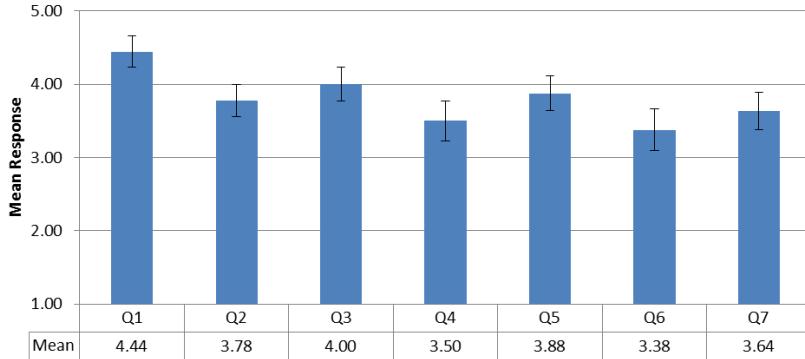


Figure 4: Mean response to the Likert scale questions and standard error bars. The scale goes from 1, completely disagree to 5, completely agree. The statements are "the following activity contributed to my learning": Q1. Creating project 4; Q2. Presenting the final demo in front of classmates; Q3. The content of the projects from other schools; Q4. Interacting with audience; Q5. Working on the project between the final demo and the C-Awards; Q6. Preparing the two-page document and the five-minute video; and Q7. Presenting project 4 at the C-Awards. Q1 and Q2 were pre-survey; Through ANOVA Multiple Measures we determine a statistically significant difference at 95% confidence only between Q1 and Q6.

the five-point Likert scale questions deployed to the eighteen students. Q1 and Q2 were part of the pre-survey deployed immediately after the final in-class presentation. The other questions were part of the post-survey, deployed immediately after the C-Awards. The scale goes from 1, completely disagree to 5, completely agree. The statements are "the following activity contributed to my learning of information visualization": Q1. Designing and creating project 4; Q2. Presenting the final demo of project 4 in front of your classmates; Q3. The content of the projects competing from other schools; Q4. Interacting with presenters from other schools; Q5. Working on the project between the final in-class demo and the C-Awards; Q6. Preparing the two-page document and the five-minute video for the C-Awards; and Q7. Presenting project 4 at the C-Awards. Through ANOVA Multiple Measures we determine a statistically significant difference at 95% confidence only between Q1 and Q6. Therefore, we can conclude that the creation of the video and two-page document for the C-Awards was not an effective mechanism to forward the ILOs in the course. We can also conclude that the participation at the C-Awards promoted the ILOs as much as the traditional activities of creating and presenting the projects in front of classmates. We expected higher results from participation at the public competition.

Next, we analyze the response to the open-ended questions. To the question "Including the C-Awards, the three best things about this course are:", 8/18 students replied that the C-Awards were among the top three elements of the course. When asked what they would change about the course, no one proposed removing the C-Awards. When asked what they learned from presenting at the C-Awards, most students highlighted their newly acquired communication skills.

6. Conclusion and Future Work

We defined Expo-Based Learning (EBL) as constructively aligning large public presentations with the methodology of project based learning. From the statistical analysis of the Likert scale question and the qualitative analysis of the open-ended questions, we conclude that participating at the open student competition, the C-Awards, presented several opportunities to promote the intended learning outcomes of the course. We interviewed the students asking for their perspective of the public element of EBL. Our next step is to conduct a qualitative analysis of the interview and reflect back on the results presented here. Through the open-ended questions we have learned so far that the greatest effect of EBL is motivating the students to improve their professionalism by delivering quality results on non-negotiable deadlines with the added pressure of presenting in front of large audiences.

7. Acknowledgements

We thank Böjrn Thuresson and Henrik Edlund for the great work managing KTH's Visualization Studio. We thank the students who participated in the information visualization course of 2014, and the organizers of the C-Awards.

References

- [RTP14] ROMERO, M., THURESSON, B., PETERS, C., KIS, P., COPPARD, J., ANDREE, J., LANDAZURI, N. Augmenting PBL with large public presentations: a case study in interactive graphics pedagogy. In *Proceedings of the 2014 conference on Innovation & technology in computer science education (ITiCSE '14)*. ACM, New York, NY, USA, (2014): 15-20.

The Effects of Peripheral Use on Video Game Play

K. Stinson and N. Bohman

Södertörns högskola, Institutionen för naturvetenskap, miljö och teknik

Abstract

Fourteen volunteers were asked to participate in an experiment, along with answering a survey, to evaluate the performance of three peripherals: the Xbox 360 Wired Controller, a keyboard, and the Rock Band Fender Stratocaster Wired Guitar Controller. The participants played a prototype made in Unity, and their accuracy scores were analyzed in R using ANOVA. However, no significant quantifiable difference was found based on which peripheral was being used. The scores were also analyzed using Pearson's Product-Moment correlation, and we were able to determine that the variation in accuracy scores was directly linked to the participant's specific test run in the experiment. Taking this into consideration along with results of our observational data and participant feedback, we found that there were more factors at play, in regards to playability and accuracy, than just the input device itself. The learning effect of repetitive play of the prototype and input devices, the control input scheme, and the participant's chosen peripheral manipulation method all had an impact.

Categories and Subject Descriptors (according to ACM CCS): I.3.1 [Computer Graphics]: Hardware Architecture—Input Devices, K.8.0 [Personal Computing]: General—Games, H.5.2 [Information Interfaces and Presentation (e.g., HCI)]: User Interfaces—Interaction Styles

1. Introduction

From the NES Zapper for *Duck Hunt* to the Konami Dance Mat for *Dance Dance Revolution Universe*, game and interaction designers have been innovating new and creative ways for people to enjoy playing games. With the popularity of rhythm music games such as *Guitar Hero* and *Rock Band*, new specialized peripherals made for use with these games have been finding their way into living rooms all over the world. While talk of Harmonix reviving the *Rock Band* franchise for the next generations of consoles has recently been circulating gaming news outlets and conventions, there are still gamers finding new uses for their aging plastic guitars until then.

1.1. Related Work

We examined several studies, not only to gain more understanding about this field of study, but also to use as inspiration for our own experimental design. Thorpe, Ma, and Oikonomou [AT11] developed a prototype that accepted eight different types of peripheral input, focusing specifically on certain types of devices that can be deemed alternative, or non-conventional. Their choices in peripherals had

a distinct influence on how their prototype and the desired control scheme were designed, as play with each of these devices needed to work as equally well as possible. Both quantitative and qualitative data was collected and analyzed, evaluating the participants' performance and their subsequent opinions on the peripherals based on their ease of use, intuitiveness and how they suited the prototype's game design. This study had a strong influence on our own experimental design.

In another study, researchers Natapov, Castellucci and MacKenzie created a prototype to be used to evaluate the performance of two peripherals: the Nintendo Classic Controller and the Nintendo Wii Remote, with a mouse as a baseline condition [NCM09]. The experiment focused on determining the technical performance of these devices in conjunction with a Fitts' point-select task, which involves target acquisition. Afterward, the participants answered a questionnaire rating the peripherals' ease of use, perceived accuracy and smoothness of operation, along with other questions gauging the comfort involved in interacting with the devices. While we found the prototype in this study to be more of a technical test compared to the more game-like prototype design we wished to implement, we found the research meth-

ods and the survey wording to also be useful sources of inspiration.

1.2. Research Question

The goal of our study is to determine: is there a quantifiable impact on playability and accuracy due to the use of certain controllers or peripherals during video game play?

We hypothesize that accuracy scores will be affected by the choice of peripheral. We also expect a learning effect through repetitive play that will also affect accuracy. However, this learning curve will be different dependent on which peripheral is being used. We also want to examine how controller mapping and other factors may have had an impact on these results.

2. Method

2.1. Experimental Design

Our goal was to implement a mixed-methods design approach to our experiment, where we gather not only quantitative data, but also qualitative. We executed an experiment with a 2×3 within-subject design. To counter the learning factor of repetitive prototype play, we introduced counterbalancing methods focusing on the testing order of each peripheral per participant. Permutations of each potential peripheral play order were found and assigned to the participants as equally as possible.

2.2. Experimental Procedure Outline

First, the sections of the questionnaire pertaining to background information were answered by the participant. Next, instructions were given describing the peripherals, their input control schemes, and the prototype gameplay mechanics. The participant was then allotted time to practice with each peripheral in a tutorial level. He or she could practice with all of the devices, or only certain ones of their choosing, for as long as they required.

First Attempt Block: The participant then played the prototype level with each peripheral in the predetermined order, resulting in the first three runs (Test Runs 1-3), where accuracy scores were recorded automatically in the prototype and/or transcribed by hand. Any pertinent feedback or comments were written down, along with observations of the participant's behavior.

After the first half of testing, the participant was asked if he or she would like to make use of the tutorial level again before proceeding to the next attempt block.

Second Attempt Block: The participant then played the same prototype level again with each peripheral in the same order as the first attempt block, resulting in the second block of three runs (Test Runs 4-6), with feedback and observations transcribed.

After the prototype testing, the participant was then asked if there was any more feedback they would like to give, especially in regards to their performance with the peripherals. The participant then completed the post-experimental evaluation section of the questionnaire.

2.3. Experimental Materials

For our experiment, the participants played the prototype on a laptop computer. The peripherals used for testing were a Xbox 360 Wired Controller, a Rock Band Stratocaster Wired Guitar Controller, and an integrated laptop keyboard.

2.4. Prototype Description and Mechanics

The prototype, made in *Unity*, features a car moving forward, bound to a four lane highway that is scattered with obstacles and colored, collectable orbs. The player can move the car in between these lanes using three different control schemes, dependent on which peripheral is being used.

We wanted a no-fail state in our test level, so collision with obstacles would only result in a decrease in accuracy scoring. The accuracy score is calculated by taking the percentage of orbs collected with deductions taken for every obstacle that was not successfully avoided.

Keyboard input is bound to the A, S, D, and F keys. This scheme was chosen based on the typing home keys for a standard keyboard, in addition to mimicking a guitar peripheral with keys in a row. Controller input is bound to the Left Trigger, Left Shoulder, Right Shoulder, and Right Trigger buttons. This scheme was chosen as it is the controller input scheme in *Guitar Hero 3* for alternative non-guitar peripheral play. Furthermore, this scheme was a viable equivalent to the row-oriented button layouts of the keyboard and guitar controllers. Guitar input is bound to the Green, Red, Yellow, and Blue buttons. This is the standard guitar peripheral scheme found in *Guitar Hero* and *Rock Band* games. Participants that have never played those games would need to learn how to use this specialized peripheral, which is why the tutorial phase of testing was allotted.

Each of these four buttons or keys is assigned to a lane on the highway, going from left to right. Movement is forced to an incremental scheme. For example, if the player is in the first lane and wants to move to the fourth, he or she must pass through the second and third lanes first. This cannot be done by simply pressing the key or button assigned to the fourth lane, so all four keys must be employed to play.

We anticipated a learning effect, not only with the peripherals, but the prototype test level as well. However, the creation of individual levels for each input device would make it difficult to compare those results with each other. So with that in mind, we had one level for use throughout the entirety of testing, and our focus was to determine exactly how much the learning effect had an impact on the results.

2.5. Participant Profile and Experiment Questionnaire

We had fourteen volunteers for experimental testing ($N = 14$), between the ages of 18 and 39. Every participant filled out a survey in conjunction with experiment testing. Basic information such as age, gender and weekly gaming habits was gathered to help form a participant profile. The participants' experience with each of the test peripherals was also inquired. Finally, the participants were asked to rate each of the three input devices based on ease of use and intuitiveness, in conjunction with their use in the prototype testing. These were rated on a seven-point Likert scale, where 1 is "Strongly Agree" and 7 is "Strongly Disagree".

2.6. Participant Feedback

Participants were encouraged to give feedback after the experiment in regards to their performance and how they felt while playing the game, if they had not already freely done so. Rubin describes "Think-Aloud" data, or verbal protocol, as the participant giving running commentary as they perform an usability test [Rub94]. With that in mind, this type of feedback was strongly encouraged as our experiment had lots of similarities to and elements of an usability test. The post-experiment interview questions were intentionally left very open ended, to avoid leading the responses given. However, if the participants needed guidance, we instructed them to discuss how interaction felt with each peripheral. Notes from this feedback were taken, in a manner as outlined in *A Practical Guide to Usability Testing*, focusing both on comments and observed behavior [DR99].

3. Results

3.1. Prototype Accuracy Score Analysis: ANOVA

The ANOVA analysis in R showed that there was no statistically significant difference between the accuracy scores based on which peripheral was being used ($p = 0.509$) nor based on which peripheral was used in conjunction with the two attempt blocks ($p = 0.782$). However, there was a highly statistically significant difference in scores themselves based on the attempt block ($p < 0.001$). We decided to analyze these scores again, focusing more on the multiple test runs done rather than the peripherals, to determine if there was a correlation.

3.2. Prototype Accuracy Score Analysis: Pearson's Product-Moment Correlation

The analysis of accuracy scores using Pearson's product-moment correlation in R showed highly statistically significant results ($p < 0.001$, $r = 0.4901$), where the play order had a positive correlation to the variation in scoring. Summarily, we can verify that 24 percent ($r^2 = 0.2401$) of the score variation is explained by the test run order (1-6), regardless of which peripheral was being used.

The results of the Pearson's analysis show that there is a strong learning effect of the test level, resulting in an independent increase of all accuracy scores towards the end of testing. This can be visualized further using a scatterplot graph, where the trend lines show that accuracy scores increased over the course of the six test runs, with all three peripherals.

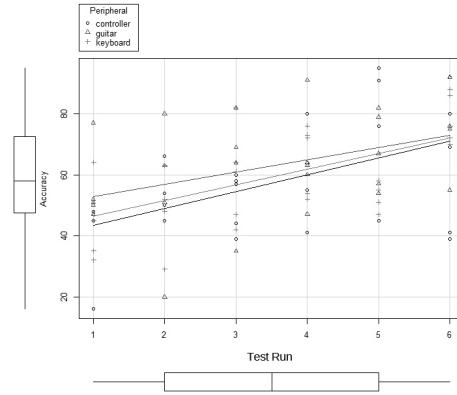


Figure 1: Scatterplot of Accuracy Scores Based on Test Run.

3.3. Participant Questionnaire

In regards to which peripheral was easiest to use, participants rated the guitar scored highest, followed by the keyboard, with the Xbox controller receiving the lowest scores (respective median values: 6, 5, 4; respective mean values: 5.5, 4.93, 3.79). Many of the participants reinforced this by remarking in the post-experiment feedback and interview that they believed they performed best with the guitar peripheral.

The guitar also scored highest when rated on intuitiveness by the participants (median value: 6, mean: 5.14), with the Xbox controller scores drastically lower in comparison (median value: 1.5, mean: 2.71). Upon use of the Xbox controller, the majority of participants immediately commented that it was the hardest to use out of the three peripherals or specifically that it had a very difficult control scheme.

3.4. Participant Observation and Feedback

No participants were "corrected" if they were using the device a certain way, even if it can be assumed that there is an intended method by design, and were allowed to use them in whatever manner felt the most comfortable. As a result, there were several instances where unique interaction methods were utilized during testing, as it was felt to be easier to manipulate the input device with a specific, unexpected technique.

The prototype's keyboard control scheme made use of the A, S, D, F keys, which can be done with the four fingers of the left hand, from pinky to index. However, we found that

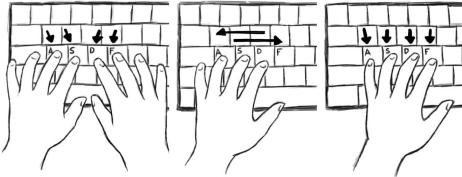


Figure 2: Observed Keyboard Interaction Methods.

certain participants felt more comfortable using their right hand. Another employed a two-handed approach, using two fingers from each hand to press the keys, resulting in relatively high scores on that device. Finally, one participant chose to only use three fingers, resting them on the standard movement keys of A, S, and D, found in many games, and shifting to the right when the F key is needed. Unsurprisingly, this participant felt that this was a very difficult way to play.

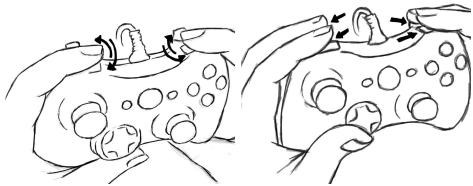


Figure 3: Observed XBox Controller Interaction Methods.

The Xbox controller, as discussed in the previous section, was deemed the most difficult to use, according to the majority of the feedback we received. Interestingly, we found that one participant had initially tried using this peripheral with only two fingers, interacting with both the shoulder and trigger buttons by shifting finger position when needed. They discovered through testing that this method was very difficult and slowed their reaction time. The participant then remarked that when they switched to using four fingers, with one on each of the controller buttons instead, their performance noticeably improved.

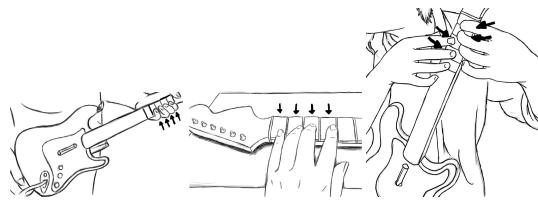


Figure 4: Observed Guitar Controller Interaction Methods.

The guitar peripheral has the most distinctive of interaction and construction designs, as it is intended to mimic an actual musical instrument. However, one participant specifically avoided holding the device like a guitar, rather opting to lay it flat on the table so that they could interact with

it like a keyboard. Another participant initially tried using the peripheral the intended way, then chose to reposition the device, interacting with the neck buttons while using two hands. The participant felt that this position was more comfortable, and subsequently felt that they had improved performance because of it.

We have found that the way in which the participants physically interacted with the peripheral had an impact on the participant's perceptions of the input device, potentially affecting their opinions on their playability.

4. Conclusions

The goal of our study was to determine if the choice of peripheral has an affect on accuracy and playability. We hypothesized that the peripheral would have an impact, but our results showed otherwise. We are unable to make a definitive conclusion in regards to accuracy based on which peripheral was used, as the results from our ANOVA analysis show no significant difference between those scores. On the other hand, we were able to show a positive correlation between accuracy scores based on which test run, out of six, the participant was playing, regardless of which peripheral was being used. Through this data, we have determined that the participants were able to improve their scores through repetitive play, even with counterbalancing measures imposed, and that practice with the peripheral and level memorization had a stronger impact on accuracy scoring.

The results of the post-experiment questionnaire showed that the participants felt that the guitar was the easiest to use and most intuitive, in comparison to the keyboard and the Xbox controller. According to the interview feedback, the Xbox controller was the most difficult to use mainly due to its input control scheme. Finally and most interestingly, we found that the intended use and design of a peripheral has no bearing at times with how it actually shall be used by the player. In conclusion, our results show that the entire interaction experience, consisting of peripherals, their input schemes, manipulation techniques, and how they're all used in tandem with a game's design, work together to influence how well a player performs and how playable it feels as a whole.

References

- [AT11] A. THORPE MINHUA MA A. O.: History and alternative game input method. In *Computer Games (CGAMES), 2011 16th International Conference on. IEEE* (2011), pp. 76–93. 1
- [DR99] DUMAS J. S., REDISH J.: *A practical guide to usability testing*. Intellect Books, 1999. 3
- [NCM09] NATAPOV D., CASTELLUCCI S., MACKENZIE I. S.: Iso 9241-9 evaluation of video game controllers. In *Proceedings of Graphics Interface 2009* (2009). 1
- [Rub94] RUBIN J.: *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*. John Wiley and Sons, Inc., 1994. 3

Playful Advertising: In-Game Advertising for Virtual Reality Games

Xiaopeng Li¹ Mario Romero²

¹ KTH Royal Institute of Technology
² HPCViz, KTH Royal Institute of Technology

Abstract

We present an early exploration of in-game advertising for virtual reality games. First, we establish a theoretical grounding for understanding interactivity and immersion in virtual reality games. Next, we report the results of a number of field studies, expert interviews, prototype designs, and describe the design of a pilot user study. Based on these results, we discuss the design of interactivity and immersion for in-game advertising and the impacts on consumer learning and game experience.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual reality

1. Introduction

Interactive 3D advertising has been receiving increasing attention and may soon become mainstream. By utilizing interactions and 3D graphics, advertisers are able to produce highly engaging and effective digital advertisements. Virtual reality, with its recent consumer-level device development, enables the interactivity and immersion of 3D graphics to be lifted to a new level. Among various types of digital advertising, in-game advertising stands out for its interactivity and immersion, which correspond to the two salient characteristics brought by virtual reality. We're set out to study in-game advertising for virtual reality games due to the shared attributes of virtual reality and games: interactivity and immersion.

Previous researches have studied the influence of interactivity on advertising [LS02] as well as the impact of 'media richness', a major antecedent of presence [LDB02]. Yet, the correlation between interactivity, immersion and sense of presence in digital advertisements is still uncharted. User experience of 3D graphics systems [YLO08] and gamers' response of in-game advertising [NKY04] have also been explored. Nevertheless, a comprehensive research framework covering both effectiveness measures and experiential metrics of in-game advertising remains missing. To fill those gaps and further explore in-game advertising in virtual reality games, we investigate the impact of interactivity and

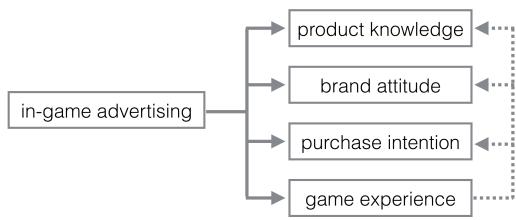
immersion of in-game advertising on consumer learning and game experience.

2. Research Framework

The study employs the measure of game experience, and three measures of the marketing effectiveness: product knowledge, brand attitude and purchase intention.

The effectiveness of 3D advertising refers to its effects on consumer learning, which can be measured by product knowledge, brand attitude, and purchase intention (e.g. [LDB02]). Interactivity is identified as one of the major factors influencing the effectiveness of interactive advertising, especially in virtual environments, with the sense of presence acting the mediating role (e.g. [LSD01, SL05]). According to Ryan [Rya99], immersion is the other main antecedent of presence beside interactivity. Studies also show that the sense of presence influences the gameplay experience (e.g. [Raa12]).

We establish a research framework, as can be seen in Figure 1, where interactivity and immersion act together to arouse a sense of presence, which further impacts the effectiveness of advertising as well as the game experience. In addition, since game experience involves game players' sensations, thoughts, feelings and actions [Raa12], we argue that it might (partially) mediate the effects of presence on the three effectiveness measures.

**Figure 1: Research Framework**

3. Field Research

The field research aims at informing the research through collecting information from academic researchers and industrial experts in the field of in-game advertising, virtual reality and human-computer interaction. We interviewed five people. Each interview lasted approximately 30 minutes. We collected data through noting and recording, and we analyzed the interview data through transcribing, coding and thematic analysis.

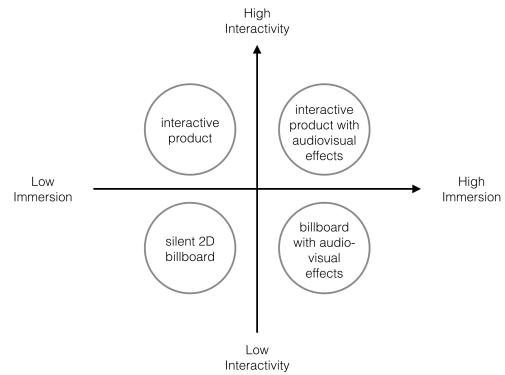
In general, interviewees held positive attitude towards in-game advertising for its potential to achieve interactivity and immersion compared to advertisements in TV or movies. They also shared the opinion that in-game advertising has the risk of hindering game experience, confirming the importance of taking the game experience into account when measuring the performance of advertising. A consensus was made among interviewees that thematic congruity between game and advertisements is crucial for game experience.

Regarding interactivity of in-game advertisements, interviewees shared the concern that interaction with advertisements may distract players from the game. However, one of the interviewees pointed out that this concern could be resolved by integrating the advertisements as part of the game. These findings threw light upon the prototype design in the following step.

Most interviewees were in favor of the application of virtual reality in media production due to its capabilities of enhancing interactivity, immersion and eventually the holistic experience. One of them also shared the perspective that positive game experience might have positive effects on gamers' attitude towards the advertised product, supporting the assumption in the research framework that game experience could influence consumer learning.

4. Prototype Design

We used a web-based virtual reality game called TRIF as the basis of the prototype. By controlling the two independent variables - interactivity and immersion, we designed four types of in-game advertisements, corresponding to four versions of prototypes. The four types of product placements are shown in Figure 2.

**Figure 2: Two Dimensions Categorizing Prototypes into Four Types**

Gamers play the game with mobile phones and Google Cardboards, which track the movement of their heads and switch the views accordingly in the game. A handled controller is used to drive the flight. Players follow a certain path in the game to collect power-ups, which are branded in certain prototypes. Along the path, they will be exposed to in-game advertisements.

5. User Study Design

A series of laboratory experiments will be conducted using between subjects design where participants are randomly divided into four groups, each group exposed to one type of the advertisements in the game. Participants will be chosen from virtual reality enthusiasts (game players, developers, researchers, and students) in Stockholm, since their prior knowledge helps alleviate the novelty effect of virtual reality and they tend to be early-adopters of future virtual reality games as well as the advertisements within.

The experiments will be set up with Moggles (counterparts of Google Cardboards) and smartphones with screen size from 4' to 5.7'. Moggles is chosen instead of Oculus Rift due to the fact that Moggles can be easily set up with smartphones enabling simultaneous execution of multiple experiments.

Participants will be welcomed with an introduction to the experiment procedures, followed by a training session where they play with ad-free version of TRIF for 5 minutes. Then, the experiment session will start and last for 10 minutes, followed by an individual questionnaire containing semantic differential and Likert-scale items, and a semi-structured group interview. The questionnaire aims to measure the dependent variables for the study, including sense of presence, product knowledge, brand attitude, purchase intention and game experience.

6. Future Work

In the current study, immersion is defined as sensory fidelity and operationalized as audiovisual effects in the experiments. Nevertheless, the definition of immersion varies among previous researches, and is confounded with sense of presence in some cases. A comprehensive or united definition of immersion awaits further investigation, in light of which the study on virtual gaming environments would possess more solid foundation.

In this research, the interaction with advertisements in the game is achieved by either looking at the billboards or collecting branded power-ups as response of head tracking and hand control. Since alternatives abound, further research might explore more interaction techniques among which eye tracking and haptic interfaces are of interest. As well, higher level of brand integration in the game will be explored which is expected to be more native and player-friendly.

References

- [LDB02] LI H., DAUGHERTY T., BIOCCHI F.: Impact of 3-d advertising on product knowledge, brand attitude, and purchase intention: The mediating role of presence. *Journal of Advertising* 31, 3 (2002), 43–57. [1](#)
- [LS02] LIU Y., SHRUM L.: What is interactivity and is it always such a good thing? implications of definition, person, and situation for the influence of interactivity on advertising effectiveness. *Journal of advertising* 31, 4 (2002), 53–64. [1](#)
- [LSD01] LOMBARD M., SNYDER-DUCH J.: Interactive advertising and presence: a framework. *Journal of Interactive Advertising* 1, 2 (2001), 56–65. [1](#)
- [NKY04] NELSON M. R., KEUM H., YAROS R. A.: Advertisment or adcreep game players' attitudes toward advertising and product placements in computer games. *Journal of Interactive Advertising* 5, 1 (2004), 3–21. [1](#)
- [Raa12] RAATIKAINEN O.: Dynamic in-game advertising in 3d digital games. *Nordicom Review* 33, 2 (2012), 93–102. [1](#)
- [Rya99] RYAN M.-L.: Immersion vs. interactivity: Virtual reality and literary theory. *SubStance* 28, 2 (1999), 110–137. [1](#)
- [SL05] SUH K.-S., LEE Y. E.: The effects of virtual reality on consumer learning: an empirical investigation. *Mis Quarterly* (2005), 673–697. [1](#)
- [YLO08] YOON S.-Y., LAFFEY J., OH H.: Understanding usability and user experience of web-based 3d graphics technology. *Intl. Journal of Human-Computer Interaction* 24, 3 (2008), 288–306. [1](#)

Triangulation painting

Max Pihlström, Anders Hast, Anders Brun

Department of Information Technology, Uppsala University, Sweden

Abstract

In this paper a dynamic image representation is proposed by combining advantages of both raster and vector graphics in the triangulation. In order for a dynamic mesh to remain a triangulation, a method which maintains integrity of representation is devised. Together with techniques for synthesizing paint, the end result is a configurable scheme demonstrating potential as a viable alternative for digital painting and imaging in general.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

1. Introduction

In image editing there are two main types of representation: raster graphics and vector graphics. In the field of digital imaging, what in recent years has commonly been researched are sophisticated methods for simulating traditional painting media [VLVR05, BWL04, DKI*12]. In these cases, crucially the representations are all built on top of a pixel or a vector graphics framework. For example, in an algorithm for watercolor developed at Adobe, the paint goes from “watery” polygon shapes to “dried” texture buffers but is still fundamentally a *composite* of raster and vector graphics [DKI*12]. In a different approach, Orzan et al. generate graphics using heat diffusion and vector curves but the underlying elements lack the appealing neighborhood characteristic of the pixel raster [OBB*13].

In this paper, a representation for image editing and paint synthesis will be presented which *combines* appealing characteristics of *both* raster and vector graphics in the triangulation. While the analogous nature between the pixel raster and the 2D triangulation has been recognized in earlier work [SP07], the focus of this paper will be to extend the triangulation as a representation for *dynamic* imaging. This will be done by deploying a *triangulation method* for maintaining integrity of representation during spatial transformation. With emphasis on robustness, this method solves the 2D analogue to the topological problem of handling multi-material surface tracking recently treated by Da et al. [DBG14]. Also, as part of paint synthesis, a method will be presented which can be seen as a 2D analogue to Pettersson’s continuous tessellation of 3D meshes [Pet14]. Finally,

techniques for contour blending and smoothing will be described, enabling for unique modes of digital imaging.

2. Edge preserving triangulation

2.1. The triangulation

A 2D *triangulation* is a straight-line graph subdividing the plane into triangles with no edge crossing another [dB-vKOS00]. Importantly, the triangulation shares the basic geometric abstraction capabilities of vector graphics as well as the neighborhood characteristics of raster graphics. Any polygon can be triangulated [dBvKOS00, p. 46] and become a subset of some larger triangulation; also, it is the case that the triangulation subdivides the plane just like the pixel raster except with arbitrary triangles instead of identical squares. This combination of properties is the motivation for making the triangulation the representation structure of this discourse, with color being an attribute of the triangle faces. We define a *contour edge* as an edge where the two triangle faces adjacent to the edge have sufficiently discerning colors (such as precisely undetectable by the eye, determined by some distance measure). A *contour* is a path of contour edges.

2.2. The problem of movement

Moving a vertex of a triangulation generally violates the planar condition of no edge intersecting another. In order to maintain a triangulation there need to be *topological changes* [AWY04], primarily in the way of rearrangement of edges. In general this problem can be solved in any number

of ways, but for our purposes there are some specific requirements that need to be met: we maintain that the only important element is contour and also that *any* composition of contour should be a possible end state. These two aspects motivates the approach for a triangulation method taken here, called *edge preserving triangulation* (EPT).

We will devise an algorithm which analytically determines the topological changes needed to rectify the disrupted state caused by moving a vertex to a new position. The approach can be explained by imagining the vertex traveling along the line from the original position to the new position. On this travel line there will be points of *discrete events* [Meh04] where the planar condition is violated. Violations are resolved in the order they appear on the travel line by “flipping” [dBvKOS00] edges (see Figure 1), where at each flip, the color attributes are also reassigned so as to preserve contour. By this process, a triangulation is progressively ensured along the travel line until the new vertex position is reached. See Figure 1.



Figure 1: An example of the EPT algorithm. The flipped edge is shown as a dashed line in the old color.

2.3. Terminology and assumptions

The algorithm assumes a *double-connected edge list* (DCEL) data structure [dBvKOS00] where the sign of the determinant of two edges of a triangle determines the orientation and direction of traversal. A triangulation contains only positive triangles. Depending on whether the determinant is strictly positive or non-negative we will say that a triangle is positive or non-negative respectively.

We will call the set of triangles incident to some vertex the *star* [Rot88] of the vertex and a triangle belonging to the star will be called a *star triangle*. The *hull* of a star of a vertex is the union of those edges of the star triangles not incident to the vertex; the *hull edge* of a star triangle is the edge of the triangle which intersects the hull; the *hull line* is the line which extends some hull edge infinitely in both directions. In the context of two star triangles we will say that their hull is *convex*, *concave* or *flat* referring to the angle between the two hull edges of the triangles.

Let T be a triangulation of the point set P . Let the input v be a vertex of T with the position p_0 , and let the input p_1 be the position where v is to be moved. We assume that p_1 is in the interior of the convex hull of P . For the time being we will also assume that $p_1 \notin P$, that is, the vertex is never moved so that it overlaps another.

Let us define the line segment

$$P : p(t) = p_0 + t(p_1 - p_0), t \in [0, 1]$$

where we observe that $p(0) = p_0$ and $p(1) = p_1$. Let us also associate the position of v with $p(t)$ so that t decides the state of the mesh where v is moving along P .

2.4. Finding critical points

Moving a vertex of T generally results in one or more *planar violations*, which is to say, a star triangle being non-positive. We assert without proof that moving a vertex always results in at least one positive star triangle. Since all star triangles of v are assumed to be positive at $t = 0$, we have that if some star triangle A is non-positive at $t = 1$ there must exist some $t_e \in [0, 1]$ when A is zero and the planar violation occurs. Finding t_e can be reduced to the task of determining when P intersects the hull line of A . This is expressed in the equation

$$t_e = 1 - \frac{x[h_b - h_a]y[p_1 - h_a] - y[h_b - h_a]x[p_1 - h_a]}{x[h_b - h_a]y[p_1 - p_0] - y[h_b - h_a]x[p_1 - p_0]} \quad (1)$$

where h_a and h_b are the incident vertices of the hull edge of A and x and y is the scalar components of the vector denoted in the index.

Let E be the set of tuples of all non-positive star triangles of v with the corresponding value t_e . We will call E the set of *violation events*. Let E' be the subset of E containing the violation events with the smallest value t_e , denoted by t'_e . E' may contain several violation events since these may be simultaneously occurring. We have that $p(t'_e)$ represents the mesh state when all star triangles are either positive or zero.

2.5. Resolving planar violations

In the violation event of A there are three ways the hull edge of A can be located in relation to the *half-planes* [dBvKOS00] defined by the travel line: the hull edge is on the left half-plane, on the right half-plane, or both. We will call violation events corresponding to these cases, respectively, a *left event*, a *right event*, or a *middle event*. These are all resolved by flipping.

2.5.1. The middle event

A planar violation of a middle event, occurring at t'_e , is resolved by flipping the intersected hull edge, resulting in two new star triangles. Any of the two triangles may be non-positive at $t = 1$. We assert that the violation events associated with such a triangle must occur at some t_e larger than t'_e .

2.5.2. The left and right events

Let A' be a triangle of some left event in E' – the right event is (mostly) symmetrical. We know that the hull edge of A' and the star triangle B' left-adjacent to A' is always non-concave. If it was concave, the hull line of B' would need to have been intersected by P at some earlier t , which contradicts the minimality of t'_e . Given that the hull of A' and B'

is convex, a flip between A and B resolves the planar violation of A ; importantly, the resulting non-star triangle will be positive. We assert without proof that if, as a result, the new star triangle violates the planar condition at $t = 1$, the violation occurs at some t_e greater than t'_e .

A degenerate case occurs when the hull of A' and B' is flat. Then B' is also associated with a left event for which the planar violation needs to be resolved before the violation of A' can be resolved. B' could in turn also have a left adjacent star triangle C' for which again the hull of B' and C' is flat, and this could go on. See Figure 2. But as previously asserted we know that a star always has at least one positive triangle, which means that in the chain of flat hulls there must exist a triangle Ω' sharing a convex hull with an adjacent left triangle, which means that the planar violation of Ω' can be resolved by a flip, whereupon the entire chain can be resolved.



Figure 2: Stages of a degenerate violation event.

2.5.3. Termination

Once all planar violations of E' have been resolved, $p(t'_e)$ represents a triangulated state. We can therefore imagine setting $p_0 := p(t'_e)$ and regarding the travel line distance as a *bound function*. The algorithm terminates when E is empty.

2.6. Implementation considerations

In an implementation it cannot be assumed as we did earlier that $p_1 \notin P$. The case when $p_1 \in P$ needs to be checked for continuously and is resolved by removing and re-linking pairs of triangles in the DCEL. Vertices moved outside of the hull or on its boundary need also be taken into special consideration. Finally, the color of the triangle faces need to be reassigned as seen in Figure 1. It turns out that the reassignment logic only has to adapt for each type of violation event for it to be consistent.

2.7. Properties

The following are noteworthy properties of the EPT algorithm.

- By minimal disruption to topology, contour is well-preserved. See Figure 4.
- Any triangulation is a possible end state of the algorithm.
- Any linear motion of a vertex can be collapsed into one execution of the algorithm.
- The algorithm can be made *robust*, avoiding round-off errors potentially leading to a corrupted state of the mesh.

3. Paint synthesis

3.1. Refraction

A method for introducing geometric detail in the triangulation is the insertion of an extra vertex in the middle of contour edges that are above a certain threshold length. We will call this method *refraction*. Refraction (together with EPT) produces smooth contours, particularly during simple spatial transformations such as rotation or movements induced by strokes with a pointing device. See Figure 4.

3.2. Contour blending

To achieve contour blending we consider color exchange between two triangles adjacent to a contour edge. To avoid coloring of long-spanning triangles when blending over a contour edge a simple method is to split the triangle at one of the other two edges at some maximum depth length. See Figure 3.

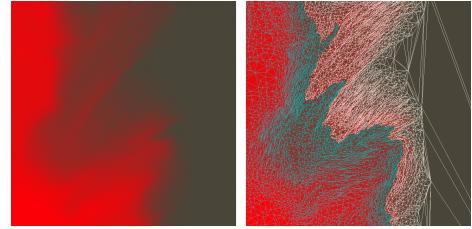


Figure 3: Contour blending produces a mesh of triangles.

3.3. Smoothing

It is possible to do *smoothing* on a region of the triangulation with a simple rule: for each vertex, calculate the average of the vectors of all contour edges and apply it as a translation vector for the vertex. When iteratively applying this rule, curve geometry is accentuated. See Figure 4.



Figure 4: A graphic and a smoothed version of it.

4. Discussion

4.1. Performance

Proof of concept software was developed to produce the results presented in this paper and was tested on a laptop

computer with a 1.86GHz Intel Core2Duo and an Intel integrated graphics card. Graphics rendering was performed on the GPU using OpenGL while other methods such as EPT were performed on the CPU in C++ code. With reasonable brush sizes, real time speed was possible with operations such as contour blending and smoothing with the smallest triangles being roughly the size of the pixels of the screen raster. Tests suggest that the EPT algorithm is the main performance bottleneck, where the computation times are proportional to the number of vertices processed and the length of their movement vectors. Since the algorithm relies heavily on vector calculations potentially a lot of performance gains can be made with parallel processing on the GPU.

4.2. Characteristics

The markedly special property of the triangulation as structure for image representation is that of contours and their spatial relationship being *integral to the structure*. This should be contrasted to raster graphics where the geometry is approximated by a grid, and vector graphics where the geometry is free but elements lack spatial relationship in terms of neighborhood. The implications of this type of integral representation is epitomized by the smoothing rule: structures can be accentuated because the total geometry is defined immediately in the triangulation with minimal compromise and redundancy. The possibilities of such rules reach beyond what has been presented here.

4.3. Future work

Currently EPT is used in research on decimating triangulation meshes in conjunction with edge-preserving filtering of images of hand-written text. The resulting meshes capture semantic content related to edge structure. See Figure 5.

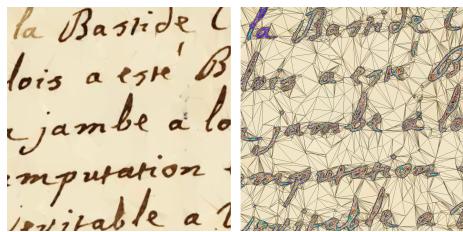


Figure 5: EPT can be used in mesh decimation.

5. Conclusion

The triangulation proves viable as a structure for dynamic representation by combining the idea of the neighborhood of raster graphics and the idea of geometric abstraction of vector graphics. In order to maintain a triangulation during transformations, a triangulation method is needed. The edge preserving triangulation (EPT) algorithm analytically

determines required topological changes that preserve contour. On top of this, methods for producing geometric detail and blending of contour can be applied along rules such as smoothing for accentuating certain aspects of geometry.

The software performs in real time for image transformations of high detail and also shows potential for further improvements in this regard. Integral representation of total geometry shows promise in providing new types of image transformations. EPT has also recently shown potential as a tool for image processing in decimating triangulation meshes.

6. Acknowledgements

I acknowledge the financial support for preparing and presenting this research from the project Searching and datamining in Large Collections of Historical Handwritten Documents (Vetenskapsrådet, Dnr 2012-5743), which is a part of the From Quill to Bytes (q2b) effort at Uppsala University.

References

- [AWY04] AGARWAL P. K., WANG Y., YU H.: A 2d kinetic triangulation with near-quadratic topological changes. In *Proceedings of the twentieth annual symposium on Computational geometry* (2004), ACM, pp. 180–189. [1](#)
- [BWL04] BAXTER W., WENDT J., LIN M. C.: Impasto: a realistic, interactive model for paint. In *Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering* (2004), ACM, pp. 45–148. [1](#)
- [DBG14] DA, BATTY, GRINSPUN: Multimaterial mesh-based surface tracking. *ACM Trans. on Graphics* (2014). [1](#)
- [dBvKOS00] DE BERG M., VAN KREVELD M., OVERMARS M., SCHWARZKOPF O.: *Computational Geometry: Algorithms and Applications*, second ed. Springer-Verlag, 2000. [1](#), [2](#)
- [DKI*12] DiVERDI S., KRISHNASWAMY A., ITO D., ET AL.: Painting with polygons: A procedural watercolor engine. [1](#)
- [Meh04] MEHTA D. P.: *Handbook of data structures and applications*. CRC Press, 2004. [2](#)
- [OBB*13] ORZAN A., BOUSSEAU A., BARLA P., WINNEMÖLLER H., THOLLOT J., SALESIN D.: Diffusion curves: a vector representation for smooth-shaded images. *Communications of the ACM* 56, 7 (2013), 101–108. [1](#)
- [Pet14] PETTERSSON T.: Sculptris. pixologic.com/sculptris (2014). [1](#)
- [Rot88] ROTMAN J. J.: *An introduction to algebraic topology*, vol. 119. Springer, 1988. [2](#)
- [SP07] SWAMINARAYAN S., PRASAD L.: Ravegrid: Raster to vector graphics for image data. In *SVG Open* (2007). [1](#)
- [VLVR05] VAN LAERHOVEN T., VAN REETH F.: Real-time simulation of watery paint. *Computer Animation and Virtual Worlds* 16, 3-4 (2005), 429–439. [1](#)

Real-Time Fluids – Optimizing Grid-Based Methods

Jaime Alvarez Losada^{†1}, Eike Falk Anderson¹ and Oleg Fryazinov¹

¹The National Centre for Computer Animation, Bournemouth University, United Kingdom

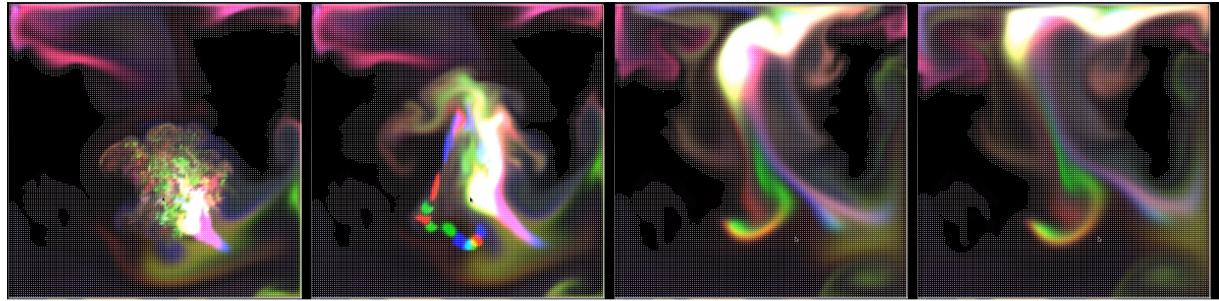


Figure 1: Several steps of a simulation, showing insignificant cells (black areas) being culled from the simulation.

Abstract

A fluid simulation suitable for use in real-time virtual environments running at interactive frame rates has the potential to greatly improve the quality of the virtual environments it is used in. To this end we present a method suitable for use in real-time grid-based fluid simulation that considerably reduces the amount of data being processed at each simulation step by removing unused cells from the simulation grid.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation; I.6.8 [Simulation and Modeling]: Types of Simulation—Visual

1. Introduction

Fluid simulation is a major research area in modern computer graphics. Most existing methods focus on offline simulations, where the quality is more important than the speed of the simulation. However, growing demand from the video games industry and interactive media has shifted attention to real-time fluid simulation methods.

Traditionally the methods to simulate fluids for real-time applications can be distinguished into two main categories: Eulerian methods, which are based on the calculation over a discretised simulation space and Lagrangian methods, which are particle-based.

In this work we are focusing on Eulerian-based methods,

which are more traditionally researched and which simplify the obtaining of effects such as compressibility of the fluids and diffusion. At the core of Eulerian methods is a fixed space subdivided by a regular grid where the parameters of the fluid (density, velocity, etc.) are calculated per cell, where one needs to find a balance between the quality of the simulation and the resolution of the grid. To increase the performance of fluid simulations based on a regular grid, we are proposing to isolate the cells that contain dynamically changing information and perform the simulation only on these cells.

The main contributions of this work are:

1. A description of the steps to reduce the number of active cells in the simulation that results in more efficient calculations;
2. An explanation of simple queries preceding the simu-

[†] first-author of this submission is a student

lation step that update the additional cells' information throughout the grid as the simulation progresses (Figure 1).

2. Background

There exist a number of physically-based approaches for fluid simulation. Compared to the popular Langrangian approaches to fluid simulation, an example of which is the recent discussion of Smoothed Particle Hydrodynamics (SPH) by Ihmsen et al. [IOS*14], the body of work related to grid-based approaches is relatively small, possibly because the overall quality of results achieved using particle-based approaches is higher. The grid-based Eulerian approach as presented by Stam [Sta03], however, lends itself particularly well to real-time simulation, as the fixed size of grid cells greatly simplifies the simulation step. This also simplifies the implementation of such simulations as GPU shaders [CLT07].

In grid-based fluid simulations a regular grid of a fixed size enclosing the simulation area is used. In this grid each cell is used to store the fluid parameters (such as velocity, density and temperature) for that particular point in space. At setup of the simulation, the grid cells are initialized to hold the initial values for the fluid parameters, and at each simulation step, changes to the cells will be calculated using the Navier-Stokes equations [Sta03].

For incompressible fluids, adaptive structures are used. Thus, Irving et al. [IGLF06] combine the cells of the simulation into tall slabs to increase efficiency and decrease memory consumption. A similar idea was used by Chentanez et al. [CM11] for water simulation on a large scale. However, the type of adaptive grid structure employed by these approaches may not be ideal for use in real-time fluid simulation, as Kallin [Kal09] suggests that a dynamically changing adaptive grid might be too computationally expensive.

For compressible fluids and smoke simulation, however, these methods are not truly suitable because of the density parameters that must be considered.

3. Method Description

Our method is based on the real-time grid-based approach presented by Stam [Sta03]. For simplicity, we present this in 2D rather than 3D, but the method itself can easily be extended to 3D.

A fluid is modelled as a velocity vector field and density scalar field which are described by Navier-Stokes equations:

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \times \nabla) \mathbf{u} + \nu \nabla^2 \mathbf{u} + \mathbf{f}$$

$$\frac{\partial p}{\partial t} = -(\mathbf{u} \times \nabla) p + k \nabla^2 p + S$$

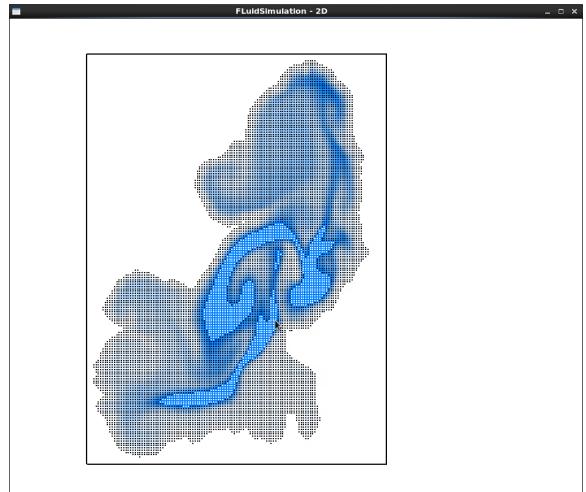


Figure 2: Cells with active surface/participation flag (black dots) denoting the simulation area, and boundary cells (dark blue outlines).

Within the data structure, grid cells contain the parameters of the fluid, such as density, pressure, and so on. We extend the data structure for grid cells by a single flag that states whether the given cell should be simulated at the given moment or whether the parameters have to be propagated from neighbouring cells. This is used as an area marker that is set for meaningful cells, i.e. those cells of the grid that contribute to the current time step, which allows the cells that do not contribute at all to be removed from the simulation. The grid size itself is not affected by this and by essentially putting aside the non-required cells, allowing the currently unused grid cells to be returned to the simulation at a later time (Figure 1), we avoid the overhead that would be caused by dynamically growing or shrinking the grid. Through an inexpensive look-up this single flag allows us to quickly retrieve information regarding the areas where the simulation actually takes place on the current frame. This in turn allows, for example, an efficient calculation of the current bounding box for the simulation.

To improve the simulation's efficiency, we perform two additional queries before each simulation step.

- On the first step we find which cells from the given regular grid participate in the simulation, for example those that carry non-zero (within a given precision) density parameters of fluid. The boundary of the grid allows us to find the bounding volume of the simulation and to process only neighbouring cells of cells isolated on the boundary during the next step of the simulation (Figure 1).
- On the second step we find the cells inside the simulation area that have identical or similar (within the given precision) parameters. Formally for two cells with velocities \mathbf{u}_1 and \mathbf{u}_2 and densities p_1 and p_2 respectively this similarity

can be defined as:

$$|1 - \mathbf{u}_1 \cdot \mathbf{u}_2| < \varepsilon_u$$

$$|p_1 - p_2| < \varepsilon_p$$

Here by ε_u and ε_p we denote tolerance parameters which define similarity. Also we assume that velocity vectors are stored as normalised.

This essentially allows us to restrict the areas for which the full simulation is required only to the boundary cells and to use the values from the boundary cells for the interior cells (Figure 2).

Finally we calculate the simulation step and update the surface flag in cells that become relevant (gain fluid) or lose relevance (no longer contain fluid) to the simulation. The very same step allows us to identify the areas with similar (within the given precision) parameters to set the participation flag for the next simulation step. Note that by varying precision we can obtain larger areas with similar parameters and therefore increase the efficiency of the simulation.

Initial results for two simulations using a 400x400 grid and running on a 64bit Linux machine with 8GB Ram and an Intel Xeon E5-1650 CPU – the first simulation implementing a conventional grid-based fluid simulation, the second adding our improvements to this simulation – showed that in the worst case (all grid cells contributing to the simulation) the performance of our method was no worse than the conventional simulation, fluctuating between 5 and 7 frames per second. In the best case (low velocity fluid, meaning that many cells could be culled) our method allowed the simulation to run about 60 times faster (445 to 440 frames per second), while on average running around 9 times faster than the conventional case (fluctuating between 45 and 60 frames per second). The same set of simulations running on a 64bit Windows machine with 4GB RAM and an Intel Core i5-2450M CPU achieved a similar performance, with the conventional case as well as our method’s worst case achieving on average 4.5 frames per second. In the best case our method achieved frame rates fluctuating between 248 and 288 frames per second, while on average framerates fluctuated between 24 and 57 frames per second.

4. Discussion and Future Work

As the evaluation of the surface/participation flag for each cell effectively culls cells that do not contribute to the simulation, in simulations where the fluid does not fill all of the simulation space there is a noticeable improvement that our method achieves in terms of processing time, when compared to existing methods. The worst case, i.e. when the fluid extends to the complete simulation area and all grid cells contribute to the simulation, meaning that the additional queries would have no effect and no longer need to be performed, our method achieves a similar performance to existing methods.

There are further improvements that we have not yet implemented. These are mainly related to which grid cells can be removed from the simulation while maintaining overall fidelity of the simulation. One has to find the best balance between speed and precision to determine setting of our simulation flag. Finding the proper balance for different cases is an area for further research.

References

- [CLT07] CRANE K., LLAMAS I., TARIQ S.: Real-Time Simulation and Rendering of 3D Fluids. In *GPU Gems 3*, Nguyen H., (Ed.). Addison-Wesley Professional, 2007, ch. 30. [2](#)
- [CM11] CHENTANEZ N., MÜLLER M.: Real-time eulerian water simulation using a restricted tall cell grid. In *ACM SIGGRAPH 2011 Papers* (2011), SIGGRAPH ’11, pp. 82:1–82:10. [2](#)
- [IGLF06] IRVING G., GUENDELMA N., LOSASSO F., FEDKIW R.: Efficient simulation of large bodies of water by coupling two and three dimensional techniques. In *ACM SIGGRAPH 2006 Papers* (2006), SIGGRAPH ’06, pp. 805–811. [2](#)
- [IOS*14] IHMSEN M., ORTHMANN J., SOLENTHALER B., KOLB A., TESCHNER M.: SPH Fluids in Computer Graphics. In *Eurographics 2014 - State of the Art Reports (STARs)* (2014), pp. 21–42. [2](#)
- [Kal09] KALLIN D.: *Real-Time Large Scale Fluids for Games*. Master’s thesis, KTH, Stockholm, 2009. [2](#)
- [Sta03] STAM J.: Real-time fluid dynamics for games. In *Game Developer Conference 2003* (2003). [2](#)

Visualizing the Effects of Public Transportation Growth on Urban Demographics

A. Bea, D. Cariño, E. Dahlström, N. Ericsson, J. Gerhardsen, E. Lagerberg, F. Stewart, and M. Romero

KTH Royal Institute of Technology, Stockholm, Sweden

Abstract

We present Tidebanan (*Time Way*), an interactive visualization of the history of the metro system of Stockholm from 1949 to 2014 that includes a projection to 2025. The analytics tool visually correlates the growth of the metro system with the changing demographic density of Stockholm. Interestingly, we discover patterns of density decrease in central Stockholm as new metro lines open, in particular the southern red and northern blue lines in the 1970s. In fact, central Stockholm has never been as dense as it was in 1949, a surprising fact that is visually salient in our application. This paper presents the design process, architecture, data models and sources, interactive visual structures, and visual analytic trails discovered through Tidebanan.

Categories and Subject Descriptors (according to ACM CCS): H.5.2 [INFORMATION INTERFACES AND PRESENTATION]: User Interfaces—Graphical user interfaces (GUI)

1. Introduction

Since its inauguration, the first subway line has molded the availability of different living areas around Stockholm. Tidebanan is a web-based visualization tool that shows the expansion of Stockholm's subway system from the year 1950 to 2025 in relation to the population growth of Stockholm's parishes (see Figure 1). At the moment, there are three filters. One filter allows the user to toggle the geographical map of Stockholm. The next one "Befolknings" shows the change in population in Stockholm's parishes as well as its bordering municipalities compared to 1950. The legend shows the values of the interval. The final filter shows population density of the these parishes and municipalities. By sliding the button on the timeline, one can move through time and see how the subway system as well as Stockholm's population has changed. One can also press on the "Play" button and the visualization automatically transitions through each year.

The motivation to create transport system visualizations is not new. Hughes provides a substantial review of the work in visualizing transport systems in geographic information systems and virtual environments that users can navigate in first person perspective and in real time [CVKA11, H04, H08, RMC04]. Tidebanan is similar to many of the systems presented there, with significant technical advancements. Primarily, it is a complete web-based application that runs on the browser, allowing for seamless cross platform operation.

Furthermore, the interactive visual structure of Tidebanan are a general framework onto which the data layers can be readily plugged in.

2. System Architecture

Tidebanan is a web-based application created in HTML and CSS frameworks. We used D3.js for subway animations, legends, and the choropleth map [BOH11]. We used JQuery for computational operations. Finally, we constructed the framework coupling the modules through javascript. We imported population data from Sweco, both present and future projections. We did not employ data query APIs for the current version of the project. Everything is done by hand, with the help of some libraries such as D3.js and jquery. The data for geographical area of the municipalities is from Wikipedia. For the computation of density, we only account for land area, not water, giving a more accurate people per livable space measurement. We map subway real-world coordinates into the visualization's coordinates. The visual structures of Tidebanan include a geographical map, a political map, a network structure accurately located on top of the geographical map, meta-layers of information represented through the colors of a choropleth map, and meta-layer information currently displaying details on demand for the individual metro stations, including history and past and present images. The view transformations include: spatial and semantic zooming,

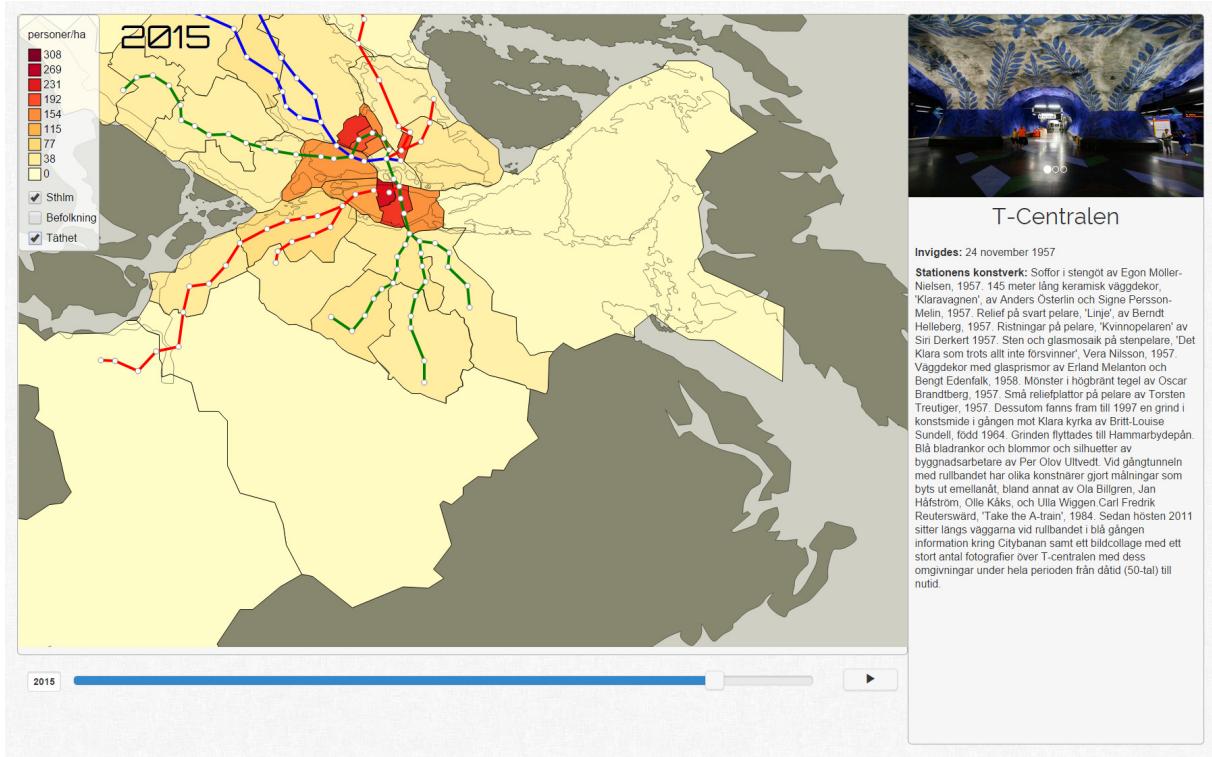


Figure 1: Tidebanan: an interactive visualization of the history of the metro system of Stockholm from 1949 to 2014 that includes a projection to 2025. The upper left corner of the main window shows the legend, next to it the year 2015. The blue timeline at the bottom also shows the year 2015. The choropleth map shows the population density of the parishes in Stockholm for the year 2015. The geographical location and identifying color of the metro lines are overlaid on the map. A gray geographical map of the city is underlaid beneath the map, highlighting the bodies of water around Stockholm. To the right of the map there is an information box where metadata about filtered objects in the map can be presented to the user. In this case, it is presenting information about the central station of Stockholm, T-Centralen. The user can zoom spatially and semantically over the map, translate, toggle information layers on and off, selected the temporal window of view, and index to details on demand regarding the individual metro stations. For a video figure of the Tidebanan, please follow the link <https://youtu.be/XMetDQAYuX4>.

panning, time travel, filtering, details on demand (clicking, hovering), real-world mapping, and selective fading.

3. Visual Analytics

Figure 2 visualizes a visual analytic trail of the evolution of Stockholm's demographics as a function of the growing metro system. Surprisingly, the population density in the center of the city is highest in 1949. When analyzed together with the growing transportation networks of the metro system, it is apparent that mobility allowed people to live further away from downtown. Interestingly, the migration pattern continues to increase dramatically farther away from the center, including the projection to 2025 on the far right.

While these insights are significant, they are limited by the layers of information the model contains. Tidebanan

presents an infrastructure that is flexible for the importing of other data types and the creation of multiple layers. Currently, the application includes historical images and information about each metro station. The metalayer information of the visualization can grow to include real-time information of traffic patterns that incorporate the bus, train, and tram systems, allowing the command and control of the transportation of a major metropolitan area in managing crises as they occur, for example.

4. Evaluation

We collected hands-on feedback from the director of the Transportation Museum in Stockholm who praised the tool for its existing affordances in opening windows of historical exploration. For him, the future of the application should include more data layers, more opportunities for commu-

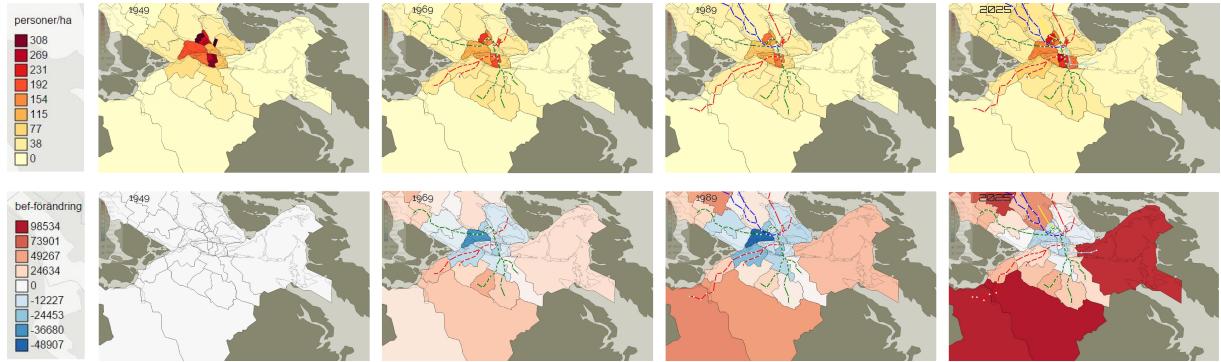


Figure 2: Analytic trail of the relationship between new metro lines and migration patterns. The first row visualizes population density with darker colors representing higher densities. The second row visualizes population migration in number of individuals. Blue represent emigration and red represents immigration in the city regions with darker shades representing higher migration numbers . The columns visualize the years 1949, 1969, 1989, and the projection to 2025. In each column we see the effects of new metro lines opening to the public. The fourth column shows the projected effects of two lines that will open before 2025. Interestingly, the highest density the center of Stockholm registered is in 1949.

nication and instruction through the tool to groups of high school students, for example, and greater hypothetical testing abilities to execute simulation scenarios. The goal for the museum is to convey the complexities of the modern urban transportation system as a fun and didactic experience for the different group ages of visitors.

5. Conclusion and Future Work

We have presented Tidebanan, a visualization of the history of the metro system of Stockholm visually correlated to the population density of the city. Tidebanan has the potential to show more layers of analysis. With the use of data query APIs, we can readily import other data types. We can build filters of income, education centers, job opportunities, prices per square meter of housing, as well as additional transportation lines such as buses and commuter trains. These filters would provide an even richer source of insight. It would be instrumental for city planners, researchers and people searching for where to make a good investment in housing. It also has the potential to be an educational tool for students and history enthusiasts.

6. Acknowledgements

We thank Böjrn Thuresson and Henrik Edlund for the great work managing KTH's Visualization Studio. We also thank Christoffer Sandahl for his evaluation.

References

- [BOH11] BOSTOCK, M., OGIEVETSKY, V., HEER, J.: D³ data-driven documents. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12), (2011): 2301-2309.

[CVKA11] CHEU, R. L., VALDEZ, M., KAMATHAM, S., ALDOURI, R.: Public preferences on the use of visualization in the public involvement process in transportation planning. *Transportation Research Record: Journal of the Transportation Research Board* 2245, 1 (2011): 17-26.

[H04] HUGHES, R. G.: Visualization in transportation: Current practice and future directions. *textit{Transportation Research Record: Journal of the Transportation Research Board}*, 1899(1), (2004): 167-174.

[H08] Hughes, R. G. Toward an Expanded Research Agenda for Visualization in Transportation: Incorporating SAFETEA-LU Directives for 'Planning'. Institute for Transportation Research and Education, North Carolina State University, Raleigh, NC (2008).

[RMC04] RAMASUBRAMANIAN, L., MCNEIL, S., CENTER, U. T. . Visualizing Urban Futures: A Review and Critical Assessment of Visualization Applications for Transportation Planning and Research. *textit{In Proceedings of the City Futures conference (2004): (9-10)}*.

Facial hair and trustworthiness in virtual faces: Towards an evaluation study

Evmorfia Kalogiannidou¹ and Christopher Peters²

¹Linkoping University, Sweden

²KTH Royal Institute of Technology, Sweden

Abstract

In this paper, we present a work-in-progress towards a perceptual study concerning male and female first impressions of male virtual faces as variations are made in terms of their facial hair (beard) length, colour and the camera viewing angle. Previous studies involving real human faces suggest that these types of variations may impact impressions of trustworthiness and related qualities, such as credibility. This research investigates similar issues, but in the domain of virtual characters, with a view to exploring issues of trustworthiness based on the appearance of virtual faces. This is of great significance to the design of interactive computational social systems.

1. Introduction

Trust plays a pivotal role in shaping human social interactions. Previous studies involving human faces have shown that impressions of trustworthiness, and related factors such as credibility, may vary based on the appearance of individuals, such as their facial characteristics [TBO08] and the angle from which the face is viewed [Sj0]. Such studies are important, not only for informing us about the way in which humans perceive their counterparts, but also for the construction of appealing interactive artificial humanoid systems, such as virtual characters and social robots [McD12]. Yet, while previous research in domains related to computer graphics and animation has focussed on rendering and animating hair, far less has considered its impact on human judgements of virtual faces, an issue of great significance to the design of interactive computational social systems.

This paper presents an overview of a preliminary perceptual study investigating the effects of facial hair and camera angle on human first impressions of trustworthiness, credibility and dominance in male virtual faces. Specifically, beard length, beard colour and camera angle are varied during the course of the study, which involves short 500ms displays of computer generated male faces. The results of the study will guide our future efforts focussing on the automatic generation, rendering and animation of facial hair capable of invoking or accounting for specific social impressions in viewers, especially in relation to trust.

2. Background

Previous research has considered trust and related concepts in both human and virtual faces. In studies involving human faces, facial characteristics such as the shape and place of the eyebrows, cheekbones, chin and nose can affect impressions of trustworthiness [TBO08]. In studies related to politics, election outcomes have been linked to candidates' perceived trustworthiness and attractiveness [LRJD12]. Studies of trustworthiness relating to facial hair and gender have led to mixed results. For example, [SP10] reports that bearded faces are less likely to be perceived as trustworthy due to relationships with facial width, while other studies [Bak14] have highlighted increased impressions of trustworthiness for bearded male faces in comparison to those without beards. A further factor that may be related to trustworthiness is camera angle. For example, in a study examining the effects of camera angle on face perception, Säfelli [Sj0] concluded that the general impression, attractiveness and the credibility of a person is altered by the vertical camera angle used. Research related to trust has also been conducted with virtual characters. Rapport agents [GWG*07], for example, engage in behavioural mimicry and backchanneling feedback in order to attempt to develop trust, liking and influence with human participants. While these studies focus on behavioural aspects, far fewer have considered the impact of appearance, and especially, varying facial hair and camera angle on impressions of trustworthiness.

3. Preliminary Study

In a preliminary study, a male face model from Facegen (<http://www.facegen.com/>) was used as a basis for a number of different experiment conditions. These included beard length variations: no beard, stubble, medium beard; colour variations: dark brown, grey; and vertical camera angles: level, high, low angle (see Figure 1).



Figure 1: Examples of stimuli used in the perceptual evaluation study. The subimages illustrate some of the conditions for beard length (top row), and beard colour and camera angle (bottom row).

The study involved 11 participants (6M:5F), aged between 21 and 29, in a within subjects design. All were students at KTH Royal Institute of Technology. Images for each of the conditions (5 trials per condition, counterbalanced) were shown to participants for a short duration (500 ms each) to probe their first impressions [WT06]. After each image, a blank screen was displayed and participants rated the trustworthiness, credibility and dominance respectively of the preceding face by selecting an option from a five point Likert scale on a written questionnaire.

4. Discussion and Future Aims

The findings of the pilot study generally suggest that alterations of the beard length, colour and the camera angle affected the perception of the virtual face. Generally, as the length of the beard increased, the virtual face was rated as more dominant but less trustworthy and credible, a result that coincides with findings in [SP10], but is at odds with the finding in [Bak14], which involved human faces. Faces displayed at low camera angles were perceived to be the least trustworthy and credible, matching findings with human faces reported in [S10]. They were also rated to be the most dominant. We intend to use this study in order to guide the design of a statistically significant user study involving a large participant population in order to compare our results with outcomes of studies involving human faces (see Section 2). There are many possibilities for future work, including accounting for the gender and culture of participants,

in addition to attractiveness, rendering style, face shape and masculinity [SP10]. In relation to gender, for example, previous studies such as [NS08] found that females rated short beards (stubble) as the most attractive option in a selection of beard types, while bearded males were rated as more masculine and aggressive than counterparts without beards. A broader aim of future perceptual studies is to inform real-time facial hair generation techniques, such as layered textured polygonal patches [Sch04], in order to allow a user to specify character appearance through high-level social perception control parameters, such as trustworthiness.

5. Acknowledgements

This work was partially supported by the European Commission (EC) and was funded by the EU Horizon 2020 ICT 644204 project ProsocialLearn. The authors are solely responsible for the content of this publication. It does not represent the opinion of the EC, and the EC is not responsible for any use that might be made of data appearing therein.

References

- [Bak14] BAKMAZIAN A.: The man behind the beard: Perception of men's trustworthiness as a function of facial hair. *Psychology* 5 (2014), 185–191. [1](#) [2](#)
- [GWG*07] GRATCH J., WANG N., GERTEN J., FAST E., DUFFY R.: Creating rapport with virtual agents. In *Intelligent Virtual Agents*, Pelachaud C., Martin J.-C., André E., Chollet G., Karpouzis K., Pelé D., (Eds.), vol. 4722 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2007, pp. 125–138. [1](#)
- [LRJD12] LITTLE A. C., ROBERTS S. C., JONES B. C., DE-BRUIJNE L. M.: The perception of attractiveness and trustworthiness in male faces affects hypothetical voting decisions differently in wartime and peacetime scenarios. *Q J Exp Psychol (Hove)* 65, 10 (2012), 2018–2032. [1](#)
- [McD12] McDONELL R.: Appealing virtual humans. In *Motion in Games*, Kallmann M., Bekris K., (Eds.), vol. 7660 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2012, pp. 102–111. [1](#)
- [NS08] NEAVE N., SHIELDS K.: The effects of facial hair manipulation on female perceptions of attractiveness, masculinity, and dominance in male faces. *Personality and Individual Differences* 45, 5 (2008), 373–377. [2](#)
- [S10] SÄTTELÉI H.-P.: The effect of different vertical camera-angles on face perception, Student thesis, University of Twente, August 2010. [1](#) [2](#)
- [Sch04] SCHEUERMANN T.: Practical real-time hair rendering and shading. In *ACM SIGGRAPH 2004 Sketches* (New York, NY, USA, 2004), SIGGRAPH '04, ACM, pp. 147–. [2](#)
- [SP10] STIRRAT M., PERRETT D.: Valid facial cues to cooperation and trust: male facial width and trustworthiness. *Psychol Sci.* (2010). [1](#) [2](#)
- [TBO08] TODOROV A., BARON S. G., OOSTERHOF N. N.: Evaluating face trustworthiness: a model based approach. *Social Cognitive and Affective Neuroscience* 3, 2 (2008), 119–127. [1](#)
- [WT06] WILLIS J., TODOROV A.: First impressions: Making up your mind after a 100 ms exposure to a face. *Psychological Science* 17, 7 (2006), 592–598. [2](#)

WOW-A-Cluster! A Visual Similarity-Based Approach to Log Exploration

J. E. Twellmeyer¹ A. Kuijper² and J. Kohlhammer^{1,2}

¹Fraunhofer IGD, Germany

²TU Darmstadt, Germany

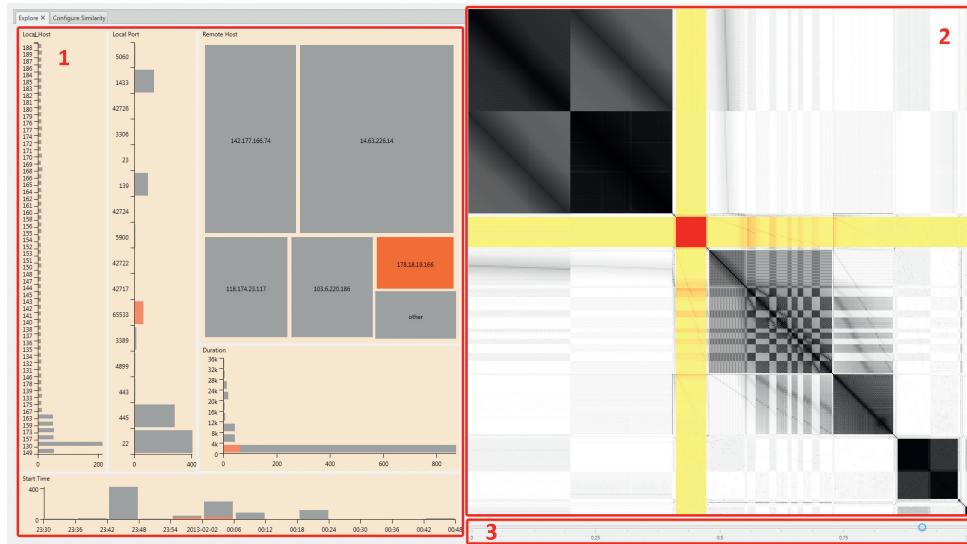


Figure 1: The WOW-A-Cluster! prototype. The log values for each entry are shown in appropriate charts on the left (1). A matrix sorted by cluster label displays the log entry similarities on the right (2). The clustering threshold can be adjusted with the slider on the bottom right(3). Brushing a cluster in the matrix causes its values in the charts to be highlighted.

Abstract

We present our work on a visual, similarity-based approach to log file exploration. The use of similarity rather than simple aggregation schemes empowers users to focus on the high-level events behind log entries, rather than the entries themselves. We make use of an accelerated version of TRIAGE to determine the similarity coefficients for each pair of log entries. The model is embedded in an interactive visualization system which enables the fluid interpretation of similarities with the help of a simple clustering approach.

Categories and Subject Descriptors (according to ACM CCS): I.5.3 [Pattern Recognition]: Clustering—Similarity measures

1. Introduction

Although logs or log files play an important role in fields, such as auditing, administration, security and forensics,

there are few visual approaches to their exploration. In this paper we present our progress on a similarity-based approach for visual log exploration.

A log is a sequence of machine-generated information entries. Examples of entries include security alerts, TCP connections and events in a process. Each entry has a time stamp and may have other important features. This definition corresponds to those given in the literature [Kre14, CS12].

Many logs are generated at a high level of granularity (such as TCP connections). Forensic analysts and network administrators are generally not interested in individual TCP connections, but rather in the high-level events which caused those TCP connections (such as a port scan). Multiple high-level events occurring simultaneously often have overlapping sets of log entries, which may obscure one another, making interpretation difficult. In order to solve this problem, state-of-the-art tools apply feature-based or index-based aggregation schemes, combined with small-multiples views on log entries. While these approaches do provide users with useful overviews, high-level events may not be revealed by simple aggregation schemes.

Thonnard et al. proposed a clustering approach to the analysis of security events called TRIAGE [TMD10, Tho10]. Clustering partitions the log into sets of entries, such that similar entries are in the same set and dissimilar entries are in different sets [KR09]. Obtaining a measure of similarity for log entries is a challenge. The TRIAGE approach assumes that features (or attributes) can be extracted from the log entries. Examples of features for a TCP connection include the source IP address, the start time and the target port. Similarity coefficients are calculated for the entries based on each feature. These similarity coefficients are then combined to form a unified similarity (see Figure 2). A key innovation of the TRIAGE approach was the use of aggregation functions, such as Ordered Weighted Averaging [Yag88] for this step. These aggregation functions enable users to integrate domain knowledge into the similarity-modeling process.

While Thonnard's approach is effective, it was not interactive; each run required expert parameterization and delivered results after minutes or hours. The aim of our prototype is to enable users in the field to apply TRIAGE to their logs on the fly. To achieve this, users must be able to configure the similarity model and receive immediate visual feedback. In addition, they must be able to interactively explore the resultant similarities in order to interpret high-level events.

In our initial prototype we have chosen to use the WOWA aggregation function [Tor96], due to its relative simplicity and effectiveness in practice. The WOWA function is parametrized with two weight vectors as shown in Figure 2. The first vector weights similarity coefficients based on their source feature. The second vector weights similarity coefficients based on their magnitude.

2. Related Work

Our presentation of related work focuses on two key areas; log file exploration and the visualization of similarities.

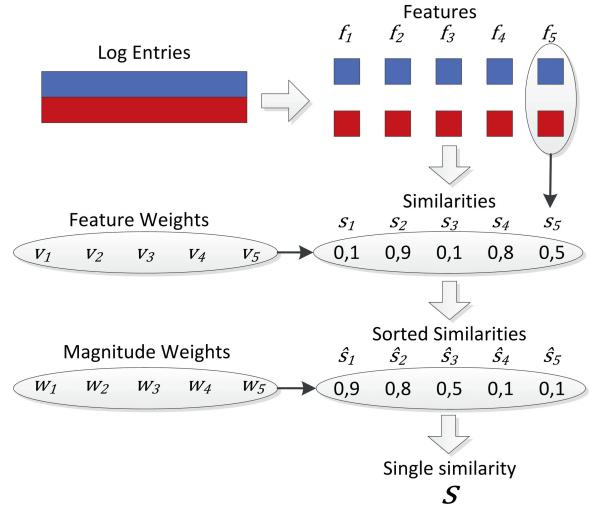


Figure 2: Schematic illustration of the components of the WOWA function.

While there is a lot of related work dealing with specific log file types, there are very few published general approaches to log file exploration. Progress has been made in specific domains, such as process mining [Aal11]. However, no general theory of log file analysis has emerged to date [CS12].

A recent system for the visual exploration of log files is ELVIS [HPBM13], which provides a set of feature types and corresponding visualizations for their exploration. The software vendor Splunk offers an index-based, rather than a feature-based approach to log file exploration [Spl]. Both ELVIS and the Splunk system include set of linked charts, as well as methods for interactive searching and filtering, and feature-based (ELVIS) or index-based (splunk) aggregation. However, neither of these approaches employ clustering to aggregate log entries into semantically meaningful groups, with respect to a multidimensional similarity model.

Numerous approaches exist for the exploration of clusters, such as parallel coordinates plots (PCPs), scatterplot matrices and sankey diagrams, which all rely on visualizing entries and their feature values. In contrast, matrices provide a view on the similarities between entries, and were proposed as a visual aid for exploratory data analysis by Jacques Bertin in 1967 [BB10]. Two prominent recent examples of matrix-based visualizations are MatrixExplorer [HFO06] and NodeTrix [HFM07]. These approaches both combine matrices with node-link diagrams; MatrixExplorer provides the user with two coordinated views on the same data and NodeTrix combines the views to a hybrid visualization. Ghoniem et al. and Keller et al. conducted user studies comparing matrices and node-link diagrams [GFC05, KEC06] and concluded that matrices were a better choice for large, dense graphs in information retrieval tasks. The usefulness of ma-

Remote host	Remote port	Local host	Local port	Start time	Duration
10.20.69.98	14067	174	3389	2013-02-01T01:31:22	1
10.214.32.177	1854	165	445	2013-02-01T01:31:23	712
10.229.0.51	43	130	42540	2013-02-01T01:31:28	1

Table 1: A sample of the dataset considered in our usage scenario

trix visualizations is highly dependent on the applied seriation (or ordering) [MML07]. CLUSION [SG03] uses a coarse seriation algorithm to provide users with a quick, compact overview of similarities with respect to a given clustering. The authors compared their approach with PCP and projection techniques to illustrate the usefulness of matrices in cluster assessment. Twellmeyer et al. presented a linked, matrix-based visualization for the exploration of security event logs clustered with the help of TRIAGE [THB*15]. The feature-based similarity matrices are displayed alongside the aggregated matrix to enable exploration. The result was an abstract view on the data and the prototype was based on static clustering results and did not enable users to modify similarity or clustering parameters.

We use a matrix as a primary means for the exploration of log entry similarities and combine this with appropriate visualizations for the feature values. Our pipeline is designed for the interactive exploration of logs; enabling parameter adjustments with almost immediate feedback.

3. Approach and Usage Scenario

In this section we present our approach and illustrate it with a typical usage scenario. For the usage scenario we used a log obtained from a HoneyPotMe [GGP] instance installed in a real network. Each entry in the log represents a malicious TCP connection with a host in the monitored network and consists of the features start time, duration (in milliseconds), the source IP address, the target host (anonymised) and the source and target ports (see Table 1).

The WOW-A-Cluster! pipeline is illustrated in Figure 3. Once the user has selected a data subset, the full pipeline is executed to display the data. The user is first presented with an overview of the data produced with the help of a default parametrization. Thereafter any changes to the parametrization of the aggregation function or clustering algorithm lead to updates. Only those parts of the pipeline are executed, which are required for the update.

The prototype has three panels; a configuration panel to configure the WOWA function, the matrix view and an exploration panel containing views of the feature values. In our usage scenario, the user configures the WOWA aggregation function in the configuration pane to reduce noise and sharpen the clusters in the data. The user then switches from the configuration to the exploration panel (see Figure 1).

The entries are clustered using graph-based clustering

methods, because the WOWA function does not guarantee that the aggregated similarity values fulfill the triangle inequality (a prerequisite for distance and density based methods). We chose the search for connected components [HT73] for our prototype, because it runs in linear time based on the number of log entries. A slider enables the user to specify the minimum aggregated similarity value for which two entries are considered connected. A variant of the coarse seriation proposed by Strehl and Gosh [SG03] is used to give the matrix its characteristic block-diagonal form. The clustering and the corresponding seriation are updated when the slider is moved. Using the threshold slider, the user is able to increase or reduce the granularity of the clustering to examine groups and subgroups.

The clusters are selectable in the matrix view, which is linked with the exploration panel. Selecting an entry cluster highlights its values in the exploration panel. By selecting salient clusters the user is able to identify clear blocks of connections from specific IP-addresses (high-level events), but also the steps of an attacker (sub-events). Examples of prominent sub-events identified in the usage scenario include port scans (an attacker testing each port of a specific host), horizontal scans (an attacker testing the same port on every host) and sustained periods of communication between the attacker and a specific host on a specific port. Examining the similarities and clusters in this way enables the user to identify the high-level events recorded in the log, and to explore the processes and phases involved in these events.

4. Conclusions & Future Work

We presented a visual, similarity-based approach to log exploration. Our prototype is an accelerated version of TRIAGE, which enables the interactive exploration of log files. The TRIAGE model is embedded in an interactive visualization system which enables the fluid interpretation of similarities with the help of clustering. We demonstrated our prototype in a usage scenario based on a real-world log. We were able to show the effective identification and interpretation of high-level events in the log.

At present it is possible to interactively explore 1000 log entries with our prototype. We are currently exploring approaches to increase this number to around 5000 entries, which would provide a reasonable compromise between the detail available to users and the level of interactivity. The WOWA function produces good results, however it requires some explanation before users can use it effectively. In ad-

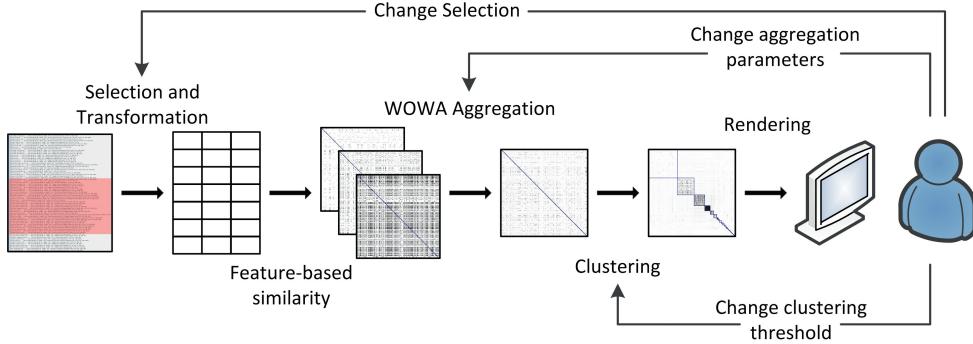


Figure 3: The WOW-A-Cluster! pipeline. A subset of the log data is selected and mapped to features. Feature-based similarities are calculated and then aggregated using the WOWA function. The data are then clustered. Users have three leverage points in the pipeline: changes to the selection, the aggregation parameters and the clustering threshold.

dition, other aggregation functions enable more flexibility in modeling entry similarity. Thus there are two further avenues of future work; integration of other aggregation functions and research into more intuitive methods of parameterizing these (e.g. through direct manipulation). Finally, we are currently acquiring a group of appropriate end users for involvement in a field study to evaluate our prototype.

References

- [Aal11] AALST W. V. D.: *Process Mining: Discovery, Conformance and Enhancement of Business Processes*, 2011 edition ed. Springer, New York, Apr. 2011. [2](#)
- [BB10] BERTIN J., BERG W. J.: *Semiology of graphics: Diagrams, networks, maps*, 1st ed ed. ESRI Press and Distributed by Ingram Publisher Services, Redlands and Calif, 2010. [2](#)
- [CS12] CHUVAKIN A. A., SCHMIDT K. J.: *Logging and Log Management: The Authoritative Guide to Understanding the Concepts Surrounding Logging and Log Management*, 1 edition ed. Syngress, Amsterdam, Dec. 2012. [2](#)
- [GFC05] GHONIEM M., FEKETE J.-D., CASTAGLIOLA P.: On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis. *Information Visualization* 4, 2 (2005), 114–135. [2](#)
- [GGP] GASSEN J., GERHARDS-PADILLA E.: HoneypotMe. https://bitbucket.org/fkie_cd_dare/honeypotme, retrieved on 17/04/2015. [3](#)
- [HF06] HENRY N., FEKETE J.: MatrixExplorer: a Dual-Representation System to Explore Social Networks. *IEEE Transactions on Visualization and Computer Graphics* 12, 5 (Sept. 2006), 677–684. [2](#)
- [HFM07] HENRY N., FEKETE J.-D., MCGUFFIN M. J.: Node-Trix: a Hybrid Visualization of Social Networks. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1302–1309. [2](#)
- [HPBM13] HUMPHRIES C., PRIGENT N., BIDAN C., MAJOR-CZYK F.: ELVIS: Extensible Log VISualization. In *Proceedings of the Tenth Workshop on Visualization for Cyber Security* (New York, NY, USA, 2013), VizSec ’13, ACM, pp. 9–16. [2](#)
- [HT73] HOPCROFT J., TARJAN R.: Algorithm 447: Efficient Algorithms for Graph Manipulation. *Commun. ACM* 16, 6 (June 1973), 372–378. [3](#)
- [KEC06] KELLER R., ECKERT C. M., CLARKSON P. J.: Matrices or node-link diagrams: which visual representation is better for visualising connectivity models? *Information Visualization* 5, 1 (2006), 62–76. [2](#)
- [KRO9] KAUFMAN L., ROUSSEEUW P. J.: *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, 2009. [2](#)
- [Kre14] KREPS J.: *I Heart Logs: Event Data, Stream Processing, and Data Integration*, 1 edition ed. O'Reilly Media, Oct. 2014. [2](#)
- [MML07] MUELLER C., MARTIN B., LUMSDAINE A.: A comparison of vertex ordering algorithms for large graph visualization. In *Asia-Pacific Symposium on Visualisation 2007* (2007), pp. 141–148. [3](#)
- [SG03] STREHL A., GHOSH J.: Relationship-Based Clustering and Visualization for High-Dimensional Data Mining. *INFORMS Journal on Computing* 15, 2 (2003), 208–230. [3](#)
- [Spl] SPLUNK INC.: Operational Intelligence, Log Management, Application Management, Enterprise Security and Compliance. <http://www.splunk.com/>, retrieved on 17/04/2015. [2](#)
- [THB*15] TWELLMAYER J., HUTTER M., BEHRISCH M., KOHLHAMMER J., SCHRECK T.: The Visual Exploration of Aggregate Similarity for Multi-dimensional Clustering. In *Proceedings of International Conference on Information Visualization Theory and Applications* (Mar. 2015), pp. 40–50. [3](#)
- [Tho10] THONNARD O.: *A Multi-Criteria Clustering Approach to Support Attack Attribution in Cyberspace*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, Paris, 2010. [2](#)
- [TMD10] THONNARD O., MEES W., DACIER M.: On a multi-criteria clustering approach for attack attribution. *ACM SIGKDD Explorations Newsletter* 12, 1 (2010), 11. [2](#)
- [Tor96] TORRA V.: Weighted OWA operators for synthesis of information. In *IEEE 5th International Fuzzy Systems* (1996), pp. 966–971. [2](#)
- [Yag88] YAGER R. R.: On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on Systems, Man, and Cybernetics* 18, 1 (1988), 183–190. [2](#)