

Proceedings of SIGRAD 2017

August 17-18, 2017 Norrköping, Sweden

Edited by Ingrid Hotz and Martin Falk

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. 143 ISSN: 1650-3686 eISSN: 1650-3740 ISBN: 978-91-7685-384-9 URL: http://www.ep.liu.se/ecp/contents.asp?issue=143

© 2017, The Authors

FOREWORD

The annual meeting 2017 of the Swedish Computer Graphics Association (SIGRAD) took place at Linköping University, Campus Norrköping in Norrköping, Sweden in August 2017. SIGRAD is an event where researchers and industry professionals meet to discuss novel visions and developments in the field of computer graphics and related areas, such as visualization and human-computer interaction (HCI). Since SIGRAD was started in 1976, it has developed into the major annual appointment for the Nordic community of graphics and visual computing experts with a broad range of backgrounds. It thereby addresses the increasing need for visual computing solutions in both commercial and academic areas. SIGRAD 2017 offered a strong scientific program consisting of international keynote speakers from research and industry, presentations of recent scientific achievements in the field within Sweden, and novel technological results from international contributors. The topics covered present a nice cross-section across the diverse research efforts in the domains.

Five original papers have been accepted for presentation after being peer-reviewed by an International Program Committee consisting of 22 highly qualified scientists. Each paper was reviewed, on average, by three reviewers from the committee. The accepted papers range from general computer graphics practices to practical applications and services that may benefit from the use of visualizations and computer graphics technologies. The extended participation of students at all levels of academia in research has been encouraged this year and 2 papers were selected which are first-authored by students studying at Master's Degree level.

This year, we continued the "Swedish Research Overview Session" introduced at last year's conference. In this session, Swedish research groups are given the opportunity to present their academically outstanding, previously published work at the annual conference. All papers in this session have been published in an academically outstanding journals or conferences not more than two years prior to the SIGRAD conference.

We especially wish to thank our invited keynote speakers: Christoph Garth, University of Kaiserslautern, Germany, Ivan Viola, Vienna University of Technology, Austria, Claes Lundström, CMIV, Linköping University, and Samuel Ranta Eskola, Microsoft. Finally, we want to express our thanks to Gun-Britt Löfgren for helping us in organizing this event.

The SIGRAD 2017 organizers

Martin Falk, Daniel Jönsson, Ingrid Hotz

TABLE OF CONTENTS

Foreword IV
Venue
Program
Organization
Keynotes IX
Swedish Research Overview SessionX
High-Quality Real-Time Depth-Image-Based-Rendering J. Ogniewski
and a 360° Video M. Tarnawski
From Visualization Research to Public Presentation – Design and Realization of a Scientific Exhibition <i>M. Krone, K. Schatz, N. Hieronymus, C. Müller, M. Becher, T. Barthelmes, A. Cooper,</i> <i>S. Currle, P. Gralka, M. Hlawatsch, L. Pietrzyk, T. Rau, G. Reina, R. Trefft and T. Ert.</i>
Evaluating the Influence of Stereoscopy on Cluster Perception in Scatterplots <i>Christian van Onzenoodt, Julian Kreiser, Dominique Heer, Timo Ropinski</i> 25
Concepts of Hybrid Data Rendering T. Gustafsson, W. Engelke, R. Englund and I. Hotz

VENUE

Linköping University, Campus Norrköping

Linköping University, LiU, conducts world-leading, boundary-crossing research in fields that include materials science, IT and hearing. In the same spirit, the university offers many innovative educational programs, frequently with a clear professional focus and leading to qualification as, for example, doctors, teachers, economists and engineers.

LiU was granted university status in 1975 and today has 27,000 students and 4,000 employees. The students are among the most desirable in the labor market and international rankings consistently place LiU as a leading global university.

Campus Norrköping is located in the city center of Norrköping in the middle of Industrilandskapet, a historical industrial area from as early as the 1750's), next to the river Motala Ström.



PROGRAM

	Thursday	Friday
	August 17	August 18
9:00 - 10:00		Paper Session I
10:00 - 10:20		Coffee Break
10:20 - 11:00		Paper Session II
11:00 - 11:45		Industrial Keynote
11:45 - 12:00		Closing Remarks
12:30 - 13:00	Registration	
13:00 - 13:15	Opening	
13:15 - 14:00	Keynote I / II	
14:00 - 15:10	Swedish Research Overview Session I	
15:10 - 15:30	Coffee Break	
15:30 - 16:15	Keynote III	
16:15 - 17:25	Swedish Research Overview Session II	
17:30 - 18:00	Mingle, Visualization Center C	
18:00 - 19:00	Dome Show	
19:00 - 20:30	Reception, Visualization Center C	

ORGANIZATION

SIGRAD Co-Chairs

Ingrid Hotz, *Linköping University, Sweden* Martin Falk, *Linköping University, Sweden* Daniel Jönsson, *Linköping University, Sweden*

International Program Committee

Eike F. Anderson, Bournemouth University, UK Nils Andersson, EON Reality, Sweden Ulf Assarsson, Chalmers University of Technology, Sweden Cornelia Auer, Potsdam Institute for Climate Impact Research, Germany Michael Doggett, Lund University, Sweden Morten Fjeld, Chalmers University, Sweden Eduard Gröller, TU Vienna, Austria Anders Hast, Uppsala University, Sweden Helwig Hauser, University of Bergen, Norway Andreas Kerren, Linnaeus University, Sweden Lars Kjelldahl, KTH Stockholm, Sweden Michael Krone, University of Stuttgart, Germany Heike Leitte, Heidelberg University, Germany Noeska Natasja, Smit University of Bergen, Norway Timo Ropinski, Ulm University, Germany Stefan Seipel, University Gävle, Sweden Örjan Smedby, KTH Royal Institute of Technology, Sweden Veronica Sundstedt, Blekinge Institute of Technology, Sweden Jonas Unger, Linköping University, Sweden Katerina Vrotsou, Linköping University, Sweden Tino Weinkauf, KTH Royal Institute of Technology, Sweden Anders Ynnerman, Linköping University, Sweden

KEYNOTES

Keynote I

Task-Based Parallelization for Visualization Algorithms Christoph Garth, *University of Kaiserslautern, Germany*

Keynote II

Visual Integration of Molecular and Cell Biology Ivan Viola, *Vienna University of Technology, Austria*

Keynote III

The role of visualization in the world of AI Claes Lundström, *CMIV*, *Linköping University*, *Sweden*

Artificial intelligence (AI), in particular deep learning, is considered to have the potential to revolutionize many domains. Even though some inflated expectations will prove unrealistic, there are many examples that clearly show how groundbreaking impact AI will have. But does visualization have a role to play in a world dominated by automated analytics? This talk will cover a few aspects of this issue in the context of applications from medical imaging diagnostics.

Industrial Keynote II

Games are Defining the Future

Samuel Ranta Eskola, Microsoft

The games industry has in a couple of decades moved from a Jolt cola-drinking basement culture to pushing technological invention all around the world. There are many examples of technologies and ideas that have been pushed forward within the games industry.

One example is Simplygon, which was spawned as a technology for the games industry. In 2017, the team joined with Microsoft in 2017 to take part in the development of 3D for everyone. We'll also look at technologies like the GPU that was pushed forward by games and now is used in cancer treatment. How VR spawned in many shapes and forms in games and now is driving car sales. Or how the Kinect was developed by game developers, now has many use cases outside of game and then later morphed into the Hololens.

We'll use our spy glass to consider how games will affect our future as well.

SWEDISH RESEARCH OVERVIEW SESSION

MVN-Reduce: Dimensionality Reduction for the Visual Analysis of Multivariate Networks

R. M. Martins, J. F. Kruiger, R. Minghim, A. C. Telea, and A. Kerren *Linnaeus University*

EuroVis 2017 (Short Paper)

Link: http://www.cs.rug.nl/~alext/PAPERS/EuroVis17/paper2.pdf

Abstract: The analysis of Multivariate Networks (MVNs) can be approached from two different perspectives: a multidimensional one, consisting of the nodes and their multiple attributes, or a relational one, consisting of the network's topology of edges. In order to be comprehensive, a visual representation of an MVN must be able to accommodate both. In this paper, we propose a novel approach for the visualization of MVNs that works by combining these two perspectives into a single unified model, which is used as input to a dimensionality reduction method. The resulting 2D embedding takes into consideration both attribute- and edge-based similarities, with a user-controlled trade-off. We demonstrate our approach by exploring two real-world data sets: a co-authorship network and an open-source software development project. The results point out that our method is able to bring forward features of MVNs that could not be easily perceived from the investigation of the individual perspectives only.

SAH guided spatial split partitioning for fast BVH construction

Per Ganestam and Michael Doggett *Lund University*

Computer Graphics Forum (Proceedings of Eurographics), Volume 35, No. 2, 2016

Link: http://fileadmin.cs.lth.se/graphics/research/papers/2016/splitting/

Abstract: We present a new SAH guided approach to subdividing triangles as the scene is coarsely partitioned into smaller sets of spatially coherent triangles. Our triangle split approach is integrated into the partitioning stage of a fast BVH construction algorithm, but may as well be used as a stand-alone pre-split pass. Our algorithm significantly reduces the number of split triangles compared to previous methods, while at the same time improving ray tracing performance compared to competing fast BVH construction techniques. We compare performance on Intel's Embree ray tracer and show that BVH construction with our splitting algorithm is always faster than Embree's pre-split construction algorithm. We also show that our algorithm builds significantly improved quality trees that deliver higher ray tracing performance. Our algorithm is implemented into Embree's open source ray tracing framework, and the source code will be released late 2015.

Global Feature Tracking and Similarity Estimation in Time-Dependent Scalar Fields

Himangshu Saikia and Tino Weinkauf KTH Royal Institute of Technology

Computer Graphics Forum (Proc. EuroVis) 35(3), June 2017

Link: http://www.csc.kth.se/~weinkauf/publications/abssaikia17b.html

Abstract: We present an algorithm for tracking regions in time-dependent scalar fields that uses global knowledge from all time steps for determining the tracks. The regions are defined using merge trees, thereby representing a hierarchical segmentation of the data in each time step.

The similarity of regions of two consecutive time steps is measured using their volumetric overlap and a histogram difference. The main ingredient of our method is a directed acyclic graph that records all relevant similarity information as follows: the regions of all time steps are the nodes of the graph, the edges represent possible short feature tracks between consecutive time steps, and the edge weights are given by the similarity of the connected regions. We compute a feature track as the global solution of a shortest path problem in the graph. We use these results to steer the - to the best of our knowledge - first algorithm for spatio-temporal feature similarity estimation. Our algorithm works for 2D and 3D time-dependent scalar fields. We compare our results to previous work, showcase its robustness to noise, and exemplify its utility using several real-world data sets.

Towards Perceptual Optimization of the Visual Design of Scatterplots

Luana Micallef, Gregorio Palmas, Antti Oulasvirta, and Tino Weinkauf *KTH Royal Institute of Technology*

IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE PacificVis) 23(6), June 2017, Received a Best Paper Honorable Mention

Link: http://www.csc.kth.se/~weinkauf/publications/absmicallef17.html

Abstract: Designing a good scatterplot can be difficult for non-experts in visualization, because they need to decide on many parameters, such as marker size and opacity, aspect ratio, color, and rendering order. This paper contributes to research exploring the use of perceptual models and quality metrics to set such parameters automatically for enhanced visual quality of a scatterplot. A key consideration in this paper is the construction of a cost function to capture several relevant aspects of the human visual system, examining a scatterplot design for some data analysis task. We show how the cost function can be used in an optimizer to search for the optimal visual design for a user's dataset and task objectives (e.g., "reliable linear correlation estimation is more important than class separation"). The approach is extensible to different analysis tasks. To test its performance in a realistic setting, we pre-calibrated it for correlation estimation, class separation, and outlier detection. The optimizer was able to produce designs that achieved a level of speed and success comparable to that of those using human-designed presets (e.g., in R or MATLAB). Case studies demonstrate that the approach can adapt a design to the data, to reveal patterns without user intervention.

A high dynamic range video codec optimized by large-scale testing

Gabriel Eilertsen, Rafał K. Mantiuk, Jonas Unger Linköping University

IEEE International Conference on Image Processing '16, 2016.

Link: http://vcl.itn.liu.se/publications/2016/EMU16/

Abstract: While a number of existing high-bit depth video compression methods can potentially encode high dynamic range (HDR) video, few of them provide this capability. In this paper, we investigate techniques for adapting HDR video for this purpose. In a largescale test on 33 HDR video sequences, we compare 2 video codecs, 4 luminance encoding techniques (transfer functions) and 3 color encoding methods, measuring quality in terms of two objective metrics, PU-MSSIM and HDR-VDP-2. From the results we design an open source HDR video encoder, optimized for the best compression performance given the techniques examined.

On local image completion using an ensemble of dictionaries

Ehsan Miandji, Jonas Unger Linköping University

IEEE International Conference on Image Processing '16, 2016.

Link: http://vcl.itn.liu.se/publications/2016/MU16/

Abstract: In this paper we consider the problem of nonlocal image completion from random measurements and using an ensemble of dictionaries. Utilizing recent advances in the field of compressed sensing, we derive conditions under which one can uniquely recover an incomplete image with overwhelming probability. The theoretical results are complemented by numerical simulations using various ensembles of analytical and training-based dictionaries.

Transfer Function Design Toolbox for Full-Color Volume Datasets

Martin Falk, Ingrid Hotz, Patric Ljung, Darren Treanor, Anders Ynnerman, Claes Lundström *Linköping University*

IEEE Pacific Visualization Symposium (PacificVis 2017), 2017

Link: http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-134851

Abstract: In this paper, we tackle the challenge of effective Transfer Function (TF) design for Direct Volume Rendering (DVR) of full-color datasets. We propose a novel TF design toolbox based on color similarity which is used to adjust opacity as well as replacing colors. We show that both CIE L*u*v* chromaticity and the chroma component of YCbCr are equally suited as underlying color space for the TF widgets. In order to maximize the area utilized in the TF editor, we renormalize the color space based on the histogram of the dataset. Thereby, colors representing a higher share of the dataset are depicted more prominently, thus providing a higher sensitivity for fine-tuning TF widgets. The applicability of our TF design toolbox is demonstrated by volume ray casting challenging full-color volume data including the visible male cryosection dataset and examples from 3D histology.

High-Quality Real-Time Depth-Image-Based-Rendering

J. Ogniewski¹

¹Linköping University, Linköping, Sweden, jenso@isy.liu.se

Abstract

With depth sensors becoming more and more common, and applications with varying viewpoints (e.g. virtual reality) becoming more and more popular, there is a growing demand for real-time depth-image-based-rendering algorithms that reach a high quality.

Starting from a quality-wise top performing depth-image-based-renderer, we develop a real-time version. Despite reaching a high quality as well, the new OpenGL-based renderer decreases runtime by (at least) 2 magnitudes. This was made possible by discovering similarities between forward-warping and mesh-based rendering, which enable us to remove the common parallelization bottleneck of competing memory access, and facilitated by the implementation of accurate yet fast algorithms for the different parts of the rendering pipeline.

We evaluated the proposed renderer using a publicly available dataset with ground-truth depth and camera data, that contains both rapid camera movements and rotations as well as complex scenes and is therefore challenging to project accurately.

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

1. Introduction

Depth-sensors become more and more common, and are integrated in more and more devices, e.g. Microsoft Kinect, Google Tango, Intel RealSense Smartphone, and HTC ONE M8. This enables new applications such as virtual reality, 360 degree video, frame interpolation in rendering [MMB97], rendering of multi-view plus depth (MVD) content for free viewpoint and 3D display [TLLG09] [Feh04], all using depth-image-based-rendering (DIBR). DIBR has been explored before (e.g. [PG10] and [YHL16]), albeit using different algorithms and different test-sequences than in this work. Here, we start with and benchmark against a renderer that was highly optimized for quality, and use the Sintel [BWSB12] datasets, which provide ground-truth values for depth and camera parameters (thus ensuring that all errors are introduced by the projection itself), as well as sequences with complex scenes and camera movement, which are challenging to project accurately.

In an earlier paper [OF17], we examined different forwardwarping methods to develop a renderer maximizing quality. This was done by creating a flexible frame-work incorporating state-of-the-art methods as well as own novel ideas, and running an exhaustive semi-supervised automatic parameter search to estimate the optimal parameter and methods. Our final algorithm is using a forward warp technique called *splatting* [Sze11], a popular choice since this leads to a high preservation of details. However, its great disadvantage is its high computational complexity, which is even made worse by the fact that it is nontrivial to parallelize.

In this paper, we develop a real-time version of our renderer, while minimizing quality loss. This was enabled by discovering and exploiting similarities between forward warping and mesh-based projection, as well as implementing efficient, accurate algorithms for the different rendering steps. The rest of the paper is organized as follows: section 2 introduces the original renderer as well as an optimized CPU version. Section 3 discusses the similarities between forward-warping and mesh-based projection as well as the different stages of the OpenGL rendering pipeline. Section 4 presents an evaluation and section 5 concludes the paper.

2. Quality optimized forward warping

In the following, we will only describe the methods which proved to be most beneficial. For a complete comparison of the different methods the reader is referred to our original paper [OF17].



Figure 1: Example projection (taken from the alley2 sequence): Top row: input image (left), mask image as used for quality measurement (right) 2nd row: projection using the CPU versions (from frame 1 to frame 25 of the sequence), original (left) and optimized (right) 3rd row: projection using the OpenGL version (from frame 1 to frame 25 of the sequence) (left), and ground-truth frame 25 of the sequence (right)

In forward warping, the points of the input frame are splatted across a neighborhood in the target frame. We call the resulting points candidate points. In many cases, several candidate points compete for the same pixel in the target frame. These are merged using agglomerative clustering [MLS14, SV14]: two candidate points will be merged if their distance in both depth and color is small enough, using initial weights based on the distance of the projected candidate point to the center of the pixel that is currently colored. The weights are summed up, to give candidates with a higher number of original points a higher weight in consecutive mergings. If another candidate point is added to the same cluster, the summed-up weight means that the same result is received as if a weighted average of all points of the cluster would have been calculated, using the initial weights. In every step, only the two points/clusters are merged that are closest to each other, and the process is stopped when this minimal distance is higher than a predetermined threshold. Then, the point/cluster is selected which is nearest to the camera; the accumulated weights are considered in the decision as well.

To counter artifacts we discovered during our work, we introduced two extensions:

- 1. *Edge suppression*: which removes anti-aliased pixels at the edge of objects, which otherwise lead to visible lines in the output frame.
- Scale adaptive kernels: we adapt the kernel size of the splatting algorithm taking local scale change into account, using a similar method as described in section 3.1.

Also, we use an internal upscale during the splatting process, of 3 in both width and length, and a Gaussian filter with overlapping neighborhoods for the downscale, see also 3.3.

Since we did an exhaustive evaluation of different methods and parameters we are confident that the final set-up lead to overall best quality results, and thus we use the same in all following implementations (if not stated otherwise) and concentrate on reducing the runtime.

2.1. CPU-optimized version

The disadvantage of the derived projection algorithm is its high computational complexity. To reduce it, we heavily optimized the code. Among other things, this included the change of all parameters to nearby values that were a power of two, as well as replacing exponential functions by functions of the form $(1 - (\frac{d}{k_s})^2)^n$, where k_s is the kernel size, d the distance (e.g. to the center of the kernel), and n an integer chosen to match the original function, and which lead to an exponent that can easily be replaced by a few multiplications. This was done e.g. for the weight calculation of the agglomerative clustering algorithm, and is demonstrated in figure 2. Also, saving the candidate points to an intermediate data structure and merging them according to agglomerative clustering is done in the same step. This simplified the computation, but leads to slightly different results, since in some cases different points are merged, or even not merged at all. While the optimized version reached a high speed-up, it is not high enough for real-time applications. Thus, we developed a OpenGL-based version.



Figure 2: Replacing one function with a similar, less computational complex one: $e^{-0.8*(\frac{d}{k_s})^2}$ (blue), $(1-(\frac{d}{k_s})^2)^6$ (red). The x-axis shows the normalized distance $\frac{d}{k_s}$. These are the functions used in the agglomerative clustering steps: the one presented in blue is used in the original renderer, the one presented in red otherwise.

3. From forward-warping to mesh-based projection

Forward-warping is extremely difficult to implement efficiently on a GPU, since it requires parallel writing to and modifying the same memory address. However, we noticed how forward warping using agglomerative clustering can be emulated by mesh-based projection:

The main reason for the high quality of forward warping lies is that several candidate points are taken into account when coloring a pixel. As discovered earlier, agglomerative clustering leads to the highest quality in forward warping, and the idea behind agglomerative clustering is to cluster candidate points together which are likely to lie in the same neighborhood of the same object. Thus, in the ideal case different clusters are derived, where each one belongs to one specific neighborhood on one specific object, and the most likely cluster is selected for the pixel in question.

Instead, a mesh-based renderer can read this neighborhood from the input texture, and calculate the final color using a Gaussian filter on this neighborhood. This filter emulates the agglomerative clustering merging process, by calculating the weights in a similar way, however only taking the distances to the center of the kernel into account. For a calculation of the color distance we would first need to determine which color the pixel is most likely to have, which is difficult to achieve accurately in a limited computation time, and therefore omitted here. Care has however to be taken that all texture values belong to the same object.

Also, both the scale-adaptive kernel and edge suppression are included naturally, the latter because points on the border between objects will not be connected by the meshing algorithm, and thus the anti-aliased color will spread in a much more limited area. However, this will also lead to more holes (as can be seen in figure 1), even if the downsampling does cancel this out to a certain degree, since only one pixel needs to be set in the neighborhood used for coloring a pixel. The rest of the missing data can be easily filled in using a simple hole-filling algorithm, e.g. *hierarchical hole-filling* [SR10]. In the following, we take a closer look at the different pipeline stages of the OpenGL version.

3.1. Meshing

Creating high quality objects and meshes from 3D depth maps has extracted a lot of attention from the research community in recent years, an example is of course [NIH*11]. However, most approaches use several depth maps for the mesh (an exception is e.g. [KPL05]), and concentrate on single objects rather than whole scenes. Here, we are interested in constructing one or several mesh(es) for the whole scene including several objects (e.g. the girl and the house in figure 1) whose number and positions are unknown from a single depth map, to allow for real-time rendering with a low latency. Also, in our application the scene may contain moving objects (see even the girl in figure 1), which is something that still has to be explored using depth map-fusion techniques. On the other hand, we only calculate the connections of the mesh rather than also refining the vertex-positions (as is often done in meshing algorithms), and assume that this is handled by an earlier depth map refining step, such as e.g. [WLC15]. Also, for reasons of computational complexity, we assume that a point may only be connected to points it is directly neighboring in the depth map. Thus, whenever the term neighborhood is used in the following, it is refering to 3x3 neighborhoods in the depth map.

The trade-off necessary in most meshing methods is trying to connect as many points belonging to the same object as possible, while creating as few connections between different objects as possible, which is demonstrated in figure 3. We



Figure 3: Wrongly connected mesh (left): connections over object boundaries (causes stripes) and missing connections (causes black spots). Our meshing algorithm was able to remove most of these artifacts (right).(detail from the temple2 sequence)

found that the following algorithm worked very well, while being comparably inexpensive to compute:

We start by creating the input vertexes for the mesh: for every value in the input depth map one point is projected to 3D-space, using the position in the depth map, the depthvalue as well as the camera parameters of the input frame. In the next step, we determine which points should be connected to which of its neighbors. We use a spheroidapproximation for that, to allow for different geometric changes in perpendicular directions. To estimate the spheroid, we calculate the difference vectors from the central point to each of its 8 nearest neighbors in the depth map, using the projected 3D positions. We select the difference vector with the smallest length (i.e. the one originating from the nearest neighbor of the central point), and calculate which of the remaining difference vectors are the most perpendicular to this difference vector, and select the two most perpendicular, taking care that they point in (approximately) opposite directions. We again select the one of these two with the smallest length. The length of this difference vector, and the length of the difference vector we selected first, are then used as the radii of the spheroid. Rather than estimating a spheroid directly, we calculate the absolute value of the dot product of each of the remaining difference vectors to the ones selected as representing the radii, and use the results as blending weights for the respective radii to derive a local radii, one for each of the remaining difference vectors. This local radius is then multiplied by a predetermined factor (2.425 was selected based on experimental results). If the resulting local radii is greater than the length of the corresponding difference vector, the neighbor used for the calculation of this difference vector is considered to be connected. Two exceptions were made in this method: 1. if one of the two radii is smaller than a predetermined factor (0.1 was selected based on experimental results) it will be set to this factor, and 2. if the radii of a neighbor is greater than the maximal depth-range found in the depth map, divided by a predetermined factor (81.25 proved to lead to good results), it will not be connected. These two selections both maximize the number of correct connections and minimize the number of false connections, see also figure 3. We save the distances of each neighbor, divided by the local radii, where the sign determines whether or not the neighbor should be connected. From the connections, edges are calculated in the next step. An edge is created if both points have positive connections. The absolute value of both connections is added up and saved; an edge is indicated by saving it as a positive value, otherwise it is saved as a negative value.

Finally, the edges are used to create the triangles used for the mesh-based rendering. For this, always 4 directly neighboring points are considered. If they are connected on at least 3 of the 4 horizontal and vertical edges, and at least one of the diagonal edges, two triangles are created connecting the two points. Out of the possible two connections, we select the one using the diagonal with the lowest (absolute) edge value. If the 4 points are only connected in one horizontal and one vertical edge, one triangle will be created if the corresponding diagonal edge is positive as well.

3.2. Agglomerative Clustering

We do the agglomerative clustering emulation in a two step approach: during the actual point projection we save the texture coordinates rather than a color. In the second step, we use the distance between the texture coordinates of the center pixel to the texture coordinates of its 8 neighbors (multiplied by the width respectively the height of the texture) to determine the scale in x- and y-direction. The respective smallest distances are used, limited to a maximal value of 2.5. The kernel size for the Gaussian filter is then determined in the following way: if any of the two coordinates is equal or smaller than 0.5, only the two nearest pixels will be used for this direction. If it is greater than 0.5 but smaller or equal to 1.5, the kernel size will be set to 3 in this direction, if it is greater it will be set to 5. Higher kernel sizes did not lead to significant increase in quality, but lead to a high increase in the runtime, in all likelihood due to the much higher demand of memory and the higher amount of memory accesses. Then, we calculate the final color by running a kernel over this neighborhood. The weights for each color is calculated using the function presented in red in figure 2, and the distance in x- and y-direction are normalized by the scale in this direction. If the distance is larger than the scale in

J. Ogniewski / High-Quality Real-Time Depth-Image-Based-Rendering



Figure 4: Example of anti-aliasing techniques. From left to right: without anti-aliasing, 2x2 upscale using 2x2 neighborhood averages for the downscale, 2x2 upscale using 3x3 Gaussian kernels for the downscale, 4x4 upscale using 4x4 neighborhood averages for the downscale; the top row is showing the complete images, the other rows zoomed-in details

one direction, the color-value is discarded. Also, we use the edges calculated earlier to discard color-values belonging to points not connected to the pixel we are currently coloring.

3.3. Downsampling

As in the CPU versions, an internal image upscaled by 3 in both width and height was used, as well as a Gaussian 5x5 kernel for downsampling. This proved to be best during our parameter estimation of the projection framework. However, instead of calculating the weights using an exponential function as used in the original CPU approach, we use predetermined power-of-2 weights for decreased computational complexity (as used in the CPU optimized version as well). We demonstrate the differences of different downsampling methods in figure 4, with the example application of full screen anti-aliasing FSAA. Anti-aliasing is a related application, where the internal upscale is applied for similar reasons as in our DIBR approach. We chose this example to emphasize differences, an thus make them more visible. In figure 4, a Gaussian kernel with overlapping neighborhoods is compared to using averages of a non-overlapping neighborhood, which is the most popular choice for FSAA due to its simple implementation. The overlapping neighborhoods introduces a slight blur, but overall leads to results with a similar visual quality as averaging methods with higher internal resolution, and which also uses more memory accesses (16 vs 9 comparing the 4x4 averaging downsample with the 3x3 Gaussian kernel) when calculating the final color in the output image. Here, the downscale has the additional advantage of filling pixels that will not be written to otherwise, since only one pixel in the neighborhood of the upscaled image needs to be set to set a pixel in the output image.

4. Evaluation

For evaluation, we compare projected images to groundtruth images to accurately measure the projection performance of the different DIBR methods. We use the Sintel testsequences [BWSB12] for that. These provide both groundtruth depth and camera poses. Access to ground-truth data is crucial to ensure that all noise and artifacts are introduced by the projection algorithms rather than by inaccurate or noisy input data. The sequences we selected were *sleeping2*, *alley2*, *temple2*, *bamboo1* as well as *mountain1* (see figure 5). We choose these sequences since they contain moderate to J. Ogniewski / High-Quality Real-Time Depth-Image-Based-Rendering



Figure 5: Selected Sintel sequences.

high camera motion, but only few moving objects, which are currently not handled by our projection. Note that some of the sequences contain highly complex scenes and are therefore difficult to project accurately. The size of the textures and depth maps used were 1024x432.

All sequences are provided with two different texture sets: clean as well as final, where the final sequences include more accurate lighting and effects such as blur, which are omitted in the clean sequences. We selected the clean sequences since the difference between different projection algorithms is more pronounced there due to a higher level of detail, which is lessened by the effects added to the final sequences. For each projection algorithm and sequence, we projected from the first and the last frame of each sequence to all other frames of the sequence, then measured the differences between the projected and the ground-truth frame in both PSNR and multi-scale SSIM [WSB03]. The results are presented in figure 7. To reach a high accuracy, we removed pixels that are occluded in the input frames as well as those containing moving objects from the measurements. This was done by using mask images, which were created beforehand. All points were projected from the input frames to each of the respective target frames. The calculaded position was rounded up and down in both the x- and y-coordinate, and the resulting 2×2 regions were set in the mask. Before that, the depth of the projected point was compared to the depth found in the depth map of the target frame for each of the 4 pixels, and only the pixels were set where the difference between the two depth-values is comparably small, to remove moving objects from the measurements. An example mask is given in figure 1, where also example images are presented from the different projection methods.

The CPU used was an Intel Xeon E5-1607 running at 3 Ghz with 8 Gbyte of memory, and the GPU used was a GeForce GTX 770 with 2 Gbyte of memory. Timing results are given in table 1. The reason why the original CPU-based renderer performs so poorly in some sequence is due to the adaptive kernel sizes. Changes in local scale (due to cameramovement or rotation) lead to large splatting kernels and thus to an inflation of candidate points that need to be considered. In the OpenGL version, the meshing step (as described in 3.1) takes up most of the time, up to 70%, followed by the rendering and the agglomerative clustering (as described in 3.2) with ca. 20% of the time. If the same depth map is to be reused, the meshing step might be omitted in consecutive frames, thus reducing the runtime drastically.

The differences in the results between the CPU versions lie

• 1	•	.1	1. 00	•		1	0 1)
mainly	1n	the	different	meroino	process	isee also	1211
manny		une	annerene	merging	process	(bee unse	,,

Sequence	Projected	CPU,	CPU,	OpenGL	
	from	original	optimized		
Sleeping 2	1	3559	999	10.9	
	49	2516	1081	10.5	
Allow 2	1	2248	918	10.0	
Alley 2	49	1713	789	9.6	
Tampla 2	1	24724	1009	9.7	
Temple 2	49	5492	898	10.4	
Bamboo 1	1	4079	949	11.2	
	49	5852	1051	11.3	
Mountain 1	1	34963	951	10.0	
	49	4166	1113	10.5	
average		8931	976	10.4	

Table 1: Timing results in ms. Average results are given for the projection to each frame of the sequence from frame 1 and frame 49 respectively.

As suspected, the OpenGL-version leads to a lesser quality in most sequences, in some cases however it performed better. The reason for this lies in the difference how the projection works. Mesh-based projection uses triangles, which can take a nearly arbitrary form in the target frame, e.g. a line segment not aligned with any of the image axes. The forward warping however always projects to a rectangle. If this rectangle contains the whole aforementioned line-segment, the connected points will project to a multitude of pixels in the target frame they are not supposed to project to, which will be punished in our paramater estimation algorithm. Therefore the two points will be connected in the mesh-based projection, but not in the forward warping. This leads to artifacts were background-objects can shine through foreground objects whose points are not connected, as demonstrated in figure 6. On the other hand, in some cases the agglomerative clustering of the CPU version might use a cluster which is not the one nearest to the camera, but has a higher number of contributing candidate points. This is not realized in our mesh-based projection approach, and would require a modification of the OpenGL pipeline, which in all likelihood would increase the runtime. However, this is probably also one of the main reasons why the CPU versions reach a higher measured quality in most sequences.



Figure 6: Artifacts due to missing connectivity (detail from the temple2 sequence): CPU version (left) with background objects shining through foreground; OpenGL version (middle): no artifacts, the partly missing foreground objects are due to occlusion in the input frame; ground-truth image (right) This is a particular difficult projection, where the camera was rotated by nearly 90 degrees between input and output frame

5. Conclusions

We developed a real-time method for DIBR, based on a computationally complex, but quality-maximized renderer. This was done by exploiting similarities between forward warping and mesh-based projection. Despite sharing these similarities, in practice they show different behavior, meaning that it might be possible to optimize the real-time renderer further, in quality as well as in runtime. Furthermore, in realworld applications depth map and camera parameters will contain noise and inaccuracies, whose effects on the projection still have to be determined.

References

- [BWSB12] BUTLER D., WULFF J., STANLEY G., BLACK M.: A naturalistic open source movie for optical flow evaluation. In *Proceedings of European Conference on Computer Vision* (2012). 1, 5
- [Feh04] FEHN C.: Depth-image-based rendering, compression and transmission for a new approach on 3D-TV. In *Stereoscopic Displays and Virtual Reality Systems* (2004), SPIE. 1
- [KPL05] KIM S.-M., PARK J.-C., LEE K. H.: Depth-image based full 3d modeling using trilinear interpolation and distance transform. In *Proceedings of the 2005 International Conference* on Augmented Tele-existence (2005), ICAT '05. 3
- [MLS14] MALL R., LANGONE R., SUYKENS J.: Agglomerative hierarchical kernel spectral data clustering. In *IEEE Symposium* on Computational Intelligence and Data Mining (2014). 2
- [MMB97] MARK W. R., MCMILLAN L., BISHOP G.: Postrendering 3d warping. In Proceedings of 1997 Symposium on Interactive 3D Graphics (1997). 1
- [NIH*11] NEWCOMBE R. A., IZADI S., HILLIGES O., MOLYNEAUX D., KIM D., DAVISON A. J., KOHI P., SHOTTON

J., HODGES S., FITZGIBBON A.: Kinectfusion: Real-time dense surface mapping and tracking. In 2011 10th IEEE International Symposium on Mixed and Augmented Reality (2011). 3

- [OF17] OGNIEWSKI J., FORSSÉN P.-E.: Pushing the limits for view prediction in video coding. In Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 4: VIS-APP, (VISIGRAPP 2017) (2017). 1
- [PG10] PALOMO C., GATTASS M.: An efficient algorithm for depth image rendering. In Proceedings of the 9th ACM SIG-GRAPH Conference on Virtual-Reality Continuum and Its Applications in Industry (2010). 1
- [SR10] SOLH M., REGIB G. A.: Hierarchical hole-filling(HHF): Depth image based rendering without depth map filtering for 3D-TV. In *IEEE International Workshop on Multimedia and Signal Processing* (2010). 3
- [SV14] SCALZO M., VELIPASALAR S.: Agglomerative clustering for feature point grouping. In *IEEE International Conference* on Image Processing (ICIP) (2014). 2
- [Sze11] SZELISKI R.: Computer Vision: Algorithms and Applications. Springer Verlag London, 2011. 1
- [TLLG09] TIAN D., LAI P.-L., LOPEZ P., GOMILA C.: View synthesis techniques for 3D video. In *Proceedings of SPIE Applications of Digital Image Processing* (2009), SPIE. 1
- [WLC15] WANG C., LIN Z., CHAN S.: Depth map restoration and upsampling for kinect v2 based on ir-depth consistency and joint adaptive kernel regression. In *IEEE International Sympo*sium onCircuits and Systems (ISCAS) (2015). 3
- [WSB03] WANG Z., SIMONCELLI E. P., BOVIK A. C.: Multiscale structural similarity for image quality assessment. In 37th IEEE Asilomar Conference on Signals, Systems and Computers (2003). 6
- [YHL16] YAO L., HAN Y., LI X.: Virtual viewpoint synthesis using cuda acceleration. In Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology (2016). 1



J. Ogniewski / High-Quality Real-Time Depth-Image-Based-Rendering

Figure 7: Measured PSNR (a), left) and MS SSIM (b), right) between the projected images and the original images of the sequences:

From top to bottom: alley, bamboo, mountain, sleeping and temple. Both projections from frame 1 (continuous lines) and from frame 49 (dashed lines) are shown, for each of the different DIBR methods.

Note that the curves are ordered according to their performance in the legend, the curves with the highest values are mentioned first.

Treating Presence as a Noun—Insights Obtained from Comparing a VE and a 360° Video

M Tarnawski[†]

Södertörn University, Sweden martina01.tarnawski@student.sh.se

Abstract

With 360° videos becoming more commercially available, more research is needed in order to evaluate how they are perceived by users. In this study we compare a low-budget computer-generated virtual environment to a low-budget 360° video viewed in VR mode. The Igroup Presence Questionnaire (IPQ), discomfort-scores and semi-structured interviews were used to investigate differences and similarities between the two environments. The most fruitful results were obtained from the interviews. The interviews highlight problematic aspects with presence, such as the difficulty of separating reality, real and realistic, which leads to a reconsideration of treating presence as a concept. The conclusions are that VR research should benefit from treating presence as a noun, the feeling of "being there" instead of a unitary concept. We also argue that presence should not by default be considered a goal of a VR experience or VR research.

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

1. Introduction

Even though presence has been evaluated and researched upon in numerous studies [LD97,FAM*00,IFR00,WHB*07], there are still many uncertainties surrounding the topic. For instance, we need to find an agreed definition of presence [TMTC03] in order to clarify what exactly is measured and thus figure out *if*, *why* and *how* it needs to be measured. Even though no agreed definition of presence has been set, research in the area of VR has almost reached an obsession of trying to achieve presence. The most basic question seem to have been lost along the way, namely "Is presence the main goal of VR?". With new technologies available today such as 360° videos being viewed in VR mode, the determinants and definitions of presence seem to be even more confusing and perhaps misleading. The aim of the study was to compare a low-budget computer-generated virtual environment to a low-budget 360° video recorded from the real world, both including an acrophobic scenario, with focus on presence and discomfort. The use of presence and problematic aspects of the term was shown to be an interesting subject to examine. The current paper thus aims at investigating if presence always is a goal for VR, and if so, how presence can

be measured across media in a way that makes the results comparable to other studies.

Dalvandi et al. [DRC11] claimed in 2011 that little research regarding level of presence in panoramic videos has been done, and it seems that is still the case. 360° videos are omnidirectional, thus allowing the user to look around in the videos which could make the experience highly immersive [RC13]. Immersion can have an impact on presence [BBA*04]. Thus, 360° videos have great potential of inducing presence. However, since presence often is discussed in relation to VR including computer-generated virtual environments (VEs), and that the existing studies targeting presence for 360° videos often include expensive material [DRC11, RC13], we need to examine whether the term and current measurements are appropriate when evaluating lowbudget 360° videos.

2. Presence

There are various definitions of the term presence: "as though they are physically immersed in the virtual environment" [GT07, p. 343], "the sense of being inside the virtual environment" [AJGMRG11, p. 504], "being there" [IFR00, SUS95]. Lombard & Ditton [LD97] examined presence by describing six conceptualisations included in the concept of presence. According to them, the main idea of

[†] Masters student

^{© 2017} The Author(s) Eurographics Proceedings © 2017 The Eurographics Association.

presence is the *perceptual illusion of nonmediation* that refers to when a user responds as if no medium were there, that the user does not acknowledge that a medium is used.

Wirth et al. [WHB^{*}07] constructed a theoretical model for the formation of *spatial presence* and argue that a model of spatial presence is the only solution to make sure that research in presence progresses. The authors claim that two critical steps are required in order to experience spatial presence. 1. The user needs to create a mental model of the situation, a Spatial situation model (SSM). 2. From this SSM, spatial presence can occur if the second level also is achieved. which is called the *medium-as-PERF-hypothesis* and refers to the user accepting the mediated environment as primary egocentric reference frame (PERF). If these two steps are achieved, it means that the users have positioned themselves in the environment and perceived the possible actions. The model includes other factors that affect the critical steps, including for instance attention allocation, higher cognitive involvement and suspension of disbelief, the users' willingness of ignoring distractions that could affect their possible wish of entertainment, such as inconsistencies.

In addition, there are various approaches that can be taken in order to measure presence, and the most common method is post-test questionnaires and rating scales [IFR00], such as the presence questionnaire (PQ) [WS98] and the Igroup Presence Questionnaire (IPQ) [SFR01]. There are clear advantages to using these types of questionnaires such as them being easy to administer and not disrupting the user's experience. However, the questionnaires are reported after the experience, which means that aspects such as variations of the level of presence during sessions cannot be detected [Ins03]. Other post-test methods targeting presence include pictorial scales [WSPW15], interviews [MAT00] and a memory test [LDAR*02].

Dalvandi et al. [DRC11] evaluated level of presence comparing VEs created using different methods of including images and videos captured from a real environment. In order to measure presence, a 6-item questionnaire including items from different sub-scales of the IPQ was used. The VE including the panoramic video was proven to be the most expensive and time-consuming to produce among the three, however it was also the one shown to induce the highest level of presence.

There are also methods that can be used during session to measure presence such as the *Continuous presence assessment* which includes a hand-held slider [IFR00], and concurrent verbal reports [TMTC03]. However, these techniques can interrupt the user's experience [IFR00, TMTC03]. Objective measurements such as postural responses [FAM*00] or physiological measures [Ins03] are also alternatives for measuring presence.

3. Methods and Material

3.1. VE and 360° video

The recorded environment simulates the experience of being located on the top of a ladder leaned against a rooftop. A similar environment was created as a computer-generated virtual environment, VE. The sound recorded in the video was also used for the VE. Both environments only allowed the action of looking around. Figures 1 and 2 show screenshots of the environments.



Figure 1: Screenshot of the 360° video used in the study.



Figure 2: Screenshot of the VE used in the study.

3.2. Measures

Juan & Perez [JP10] compared the level of presence and anxiety in a VE and an augmented reality environment. The participants were asked to rate their level of anxiety on a 10-point scale 6 times during each experience. In this study, a similar approach is taken and the participants have reported *discomfort-scores* during the session. However, the scores have only been reported 3 times during each experience since the participants only experienced each environment for approximately 1:30 minutes. By using the term *discomfort*, the scale becomes similar to methods used in previous studies [CSSS06, KDSCH12, WBBL*15].

In order to measure the participants' perceived level of presence in the 360° video and the VE, the Igroup presence questionnaire (IPQ) was used. The IPQ contains 14 items that are answered on a 7-point scale. The items are divided in: General presence (G), Spatial presence (SP), the sense of being physically present in the environment, Involvement (INV), the experienced involvement and attention directed to the environment, and Experienced realism (REAL), the subjective experience of realism in the environment [Igr16]. In order to facilitate for the participants in this study, the IPQ was translated from English to Swedish. The fact that the questionnaire has been translated in two steps and adapted to 360° videos must be taken into consideration if the results from this study are compared with other studies using the IPQ since these versions have not been tested. However, using adapted versions of existing questionnaires to better suit the study is common [LDAR*02, TMTC03, JP10, DRC11, RC13].

Semi-structured interviews were also included in order to receive a deeper knowledge about the participants' answers. The questions were inspired by previous research as well as insights from the first sessions conducted where tendencies could be noticed regarding differences in behaviour and discomfort for the two environments. All answers from the interviews except those from the follow-up questions were recorded, transcribed, and analysed using a thematic method [BC06].

3.3. Apparatus

The HMD used in order to view the environments was Spectra Optics G-01 3D VR Glasses. The headphones used were Sennheiser HD 418 which have a close-back design that blocks out much noise from the outside. The smartphone used in the experiments was a Samsung Galaxy S6 and the application used to view the VE was the 360 VR Player |Videos.

3.4. Participants

Due to the lack of research regarding differences between 360° videos viewed in VR mode and VEs, this study aimed at including people with a broad age range in order to form a foundation for future studies. 21 participants were used for the quantitative part of the study. Their age had a range of 19–72, mean age 37.8 (SD = 19.4). 11 of the 21 participants also took part in the interview (one interview was discarded due to confusion of the environments). The participants were contacted through an art school and one workplace and remaining participants were contacted through digital channels. The participants did not receive any financial reward. The inclusion criteria for participation were that they could not wear glasses during the experiment or perceive themselves as being extremely afraid of heights.

3.5. Procedure

The sessions were conducted in a room where no other people than the researcher and the participant were present. The order in which the environments were viewed was randomized and counterbalanced resulting in 10 people starting with experiencing the VE and 11 with the 360° video.

@ 2017 The Author(s) Eurographics Proceedings @ 2017 The Eurographics Association.

Each session began with a brief introduction of the session. It was explained to the participants that they could end the session at any time and that simulator sickness can occur. Before the sessions begun, each participant was also asked if they are extremely afraid of heights in order to exclude people that could find the experience too frightening. The experience began with the participant facing the roof. A discomfort-score was registered (moment 1), and the participant was then asked to look and move around. The participant was also specifically asked to look down. After approximately 30 seconds a discomfort-score was marked (moment 2), and after around 80 seconds the last discomfortscore was registered (moment 3). After the first experience, the participant filled in the first IPQ. The next environment was experienced using the same procedure as the first, and yet another IPQ was filled in afterwards. For the 10 first participants, the session ended there. The remaining 11 were requested to answer a few questions and also if they approved that the interview was recorded in order for it to be analysed later. The sessions lasted for around 20 minutes.

4. Results

4.1. IPQ and Discomfort-scores

The scores from the IPQ were analysed using paired *t*-tests. The means for each participant's answers for the different categories for the VE was calculated and compared to the means of the 360° video, thus compiling the questions belonging to the same category. The significance level was set to 0.05 in all statistical analyses. No significant differences between the VE and the 360° video were found in the categories General presence (p = 0.521) and Spatial presence (p = 0.332). However, significant differences were found in Involvement (p = 0.031) and Experienced realism (p = 0.004) where the 360° video received higher scores. The means of the discomfort-scores for the two environments were also compared using paired t-tests. It should be noted that the participants might have reported the discomfortscores looking in different directions, however, since the environments do not include any other actions than the possibility to look around, one could also assume that a general feeling of the environment was created fairly quickly. No significant differences were found among the results (moment 1: p = 0.835, moment 2: p = 0.557, moment 3: p = 0.137). Four participants reported a discomfort-score of 0 on all moments in the 360° video. Two of these four participants also indicated the same score for the VE and the other two participants indicated higher scores for the VE.

4.2. Interviews

The first interview question was inspired by a question included in a study by Juan & Perez [JP10, p. 760] that used an adapted version of a questionnaire created by Slater et al. [SUS94]: "During the experiment, did you think that you actually were in any of the environments?" Some participants directly associated the question with the factor of realism and indicated that they felt more there in the 360° video because they found the environment more real or realistic. An expression that should be highlighted among the answers is "felt like another reality". The participant felt more as if being there in the 360° video and the statement was mostly referring to that environment. What is interesting is that the participant did not say that it felt as "reality", but as *another* reality which indicates that s/he felt present in the 360° video even though it did not feel as our reality. This view could be interesting to evaluate regarding VEs. By using the words "another reality", the comparison of the VE to the real world that many people automatically seems to do, can be minimized. Marini et al. [MFGR12] suggest that the goal of a VR experience could be to make it *believ*able, rather than real, since the aim is to convey the idea that the VR experience is the real thing. They suggest that in order to achieve a believable VR, realism is not always needed and a symbolic approach can be used. When choosing the word believable, people may ask themselves "would the world look and feel like this if it existed?" and not "does this world look much like the existing world?". An item that could be included when the goal of a VE is to make it believable is: "The virtual environment/360° video felt believable", with anchors Not at all-Completely believable. Another item could be: "The environment felt like it could exist in another reality" with anchor points Felt as it could not exist at all-Felt as if it definitely could exist. The anchor points are important since some people may feel that the environment could exist in *our* reality. An item targeting this could be: "The environment felt as if it could be a part of our real world", with anchors Not at all-Definitely.

One participant mentioned a feeling of being there in both environments, but in different ways, due to the 360° video looking more real, but also having a feeling of becoming a video game character in the VE. This person also claimed having good knowledge of video/computer games and also having tried VR earlier and finding it exciting. Due to this, it might be possible that this person had a wish to be entertained in this experience as well and it might be argued that the participant could have been more willing to overlook distractions in the VE and had a greater suspension of disbelief than others. However, this participant described the appearance of the VE in a detailed way, thus having been fully aware of inconsistencies such as standing far away from the ladder. One possibility is that the person was aware of the inconsistencies but may not have compared the experience as much with our reality but more to the feeling of playing a game, which might be viewed as another reality. The same participant made a similar comment regarding the perceived realism of the environments. The participant mentioned that the 360° video was more real since it *looked* more real, however the person said that s/he felt more static in that environment. Even though the participant stated that s/he knew that the same actions were possible in both environments, it was still perceived as if more actions were possible in the VE and it thus felt real.

From a follow-up question, the participant also explained

that the feeling of being a character in a video game lead to feeling that it should not matter if s/he was to fall down the ladder since a new life would be received as in video games, and that this made it more comforting. One participant mentioned expecting movement in the VE due to the fact that it looked much like a computer game. Another participant experienced the VE as being part of a game but found it discomforting due to not knowing what actions that were possible in the environment. This participant also mentioned having very limited knowledge about video/computer games and no previous experience of VR. The one with greater knowledge was thus aware that most video/computer games offer a second chance when failing or "dying" while the other participant might not share the same view or consider it. The later mentioned participant did not feel as *being there* in the VE which could be due to the person comparing the experience to our reality and not the feeling of being inside a game. This could be an indication of the difficulty of separating the feeling of being there to the feeling of being in a place that exists in our reality. Previous experience of VR does however not automatically lead to a greater feeling of being in the VE. Two other interviewed participants had previous experience of VR and clearly stated feeling there to a greater extent in the 360° video. What people compare the VE experience with or how they perceive the environment thus seems to be truly personal.

A majority of the interviewees perceived the 360° video as undoubtedly more real. One person mentioned in the interview that s/he recognized the recorded area in the 360° video which may have affected the answers and made the participant perceive that environment as most realistic. The feeling that the environment exists or might exist in the real world is however different from the feeling of being present in the environment but seems to be difficult separating. For 360° videos that only include recordings of existing environments, it could be redundant measuring how aesthetically realistic people perceive the environment, and other aspects such as involvement and spatial presence may be more interesting measuring which can indicate whether people felt an interest for the environment and as if they were physically there. However, one participant mentioned that the perspective in the 360° video looked unreal, thus, questions regarding how real or realistic the environment looks could be included if the goal of the environment is to make it look as realistic as possible.

Some mentioned height as the main reason for feeling discomfort. Other participants experienced nausea or vertigo when viewing the VE but not as much in the 360° video. This might be due to latency in the head-tracking movement that only occurred in the VE, which was calculated in real-time. Confusion around the feeling of being able to look around was also a reason for discomfort. Another participant mentioned that s/he was afraid of falling down the ladder and that this made the person conscious about the amount of movement s/he initiated. The participant also mentioned a feeling of wanting to grab the ladder. The person felt equally about these aspects in both environments, however the participant only mentioned trying to walk and climb up the roof in the VE that was the environment tested last.

5. Discussion

5.1. Realism

According to previous research, the feeling of presence will positively be achieved when experiencing a convincing illusion of reality [PSR13]. However, in this study it became noticeable that the term *reality* can be confused with terms such as real or realistic, particularly when translated into different languages. Item number 12 in the IPQ was confusing for some of the participants: "The virtual world/360°-video felt more realistic than the real world." In the original version of the IPQ, the German word used for *realistic* is *wirklich*. The German "wirklich" does not have an equivalent in English [Igr16]. The word "wirklich", and Swedish "verklig" can have the meaning of existing (in the world) and that can be detected through senses. This definition is very different from the Swedish translation of the word "realistic" ("realistisk") which can be described as being characterized by realism and seems to be used more often when describing that something looks or feels lifelike, as it does in the real world. This can however be confusing since the English word "realistic" can refer to when something seems to exist or be happening in fact. The word *real* could be a suitable translation of "wirklich". "Wirklich" can however have other definitions than the one mentioned earlier, which may not match the Swedish translation of "real" in this context.

For the items in the original IPQ where it is asked how "real" the environment seemed to the participant, the German real was used, which in Swedish translates to "verklig". However, in item 12, the Swedish translation of the word "realistic" was used, "realistisk". The participants seemed to perceive the question differently and thought it was difficult comprehending what was asked for. One aspect that may have contributed to the confusion was the fact that one of the tested environments was a video, recorded from the real world, and it could thus seem strange asking if the environment felt more realistic than the real world. We propose that it may be beneficial to use the term *real* instead of *realistic* in contexts where the participants are asked for the overall feeling of realism in the environment and not the aesthetical appearance. The item should also be more specific in order to avoid confusion: "During the experience, the virtual environment/360° video felt more real than the environment I was physically located in." However, we suggest that this question only should be used when the purpose of a study is to achieve realism.

There was no significant difference in the scores from the VE and the 360° video on item 12 in the IPQ (p = 0.119). This could be due to that the participants were asked if the environment *felt* more realistic, and not how it looked. The English version of the IPQ asks if the world *seemed* more

realistic than the real world which is not the term used in the IPQ in this study since the Swedish translation worked better using the equivalent of "feel".

Having in mind the aspects discussed above, it is not surprising that the 360° video received higher scores in the category Experienced realism in the IPQ. No significant differences in General presence or Spatial presence were however found which could indicate that the participants in general felt that they were physically present and felt as if being there in both environments.

5.2. Defining presence

This paper suggests that presence no longer should be treated as a unitary concept and that aspects often included in previous research as a part of the concept presence should be treated separately. This is both due to the term being ambiguous but also since presence may not always be the true or only goal for VR. This study indicates that evaluating presence becomes even more problematic when comparing level of presence in 360° videos and VEs. This is mostly due to that 360° videos look realistic and people confuse the aesthetic features to the feeling of *being there*. The results also indicate that the VE affords more actions but that it may be difficult expressing this feeling. In order to further investigate this topic, we propose the following questions: "During the experience, to what extent did you try to interact with the environment in more ways than just looking around?", "Before the experience, to what extent did you believe that you were going to be able to interact with the environment you were to experience?", and "To what extent did your initiated actions receive a response by the environment?" The questions suggested above are similar to some items of the PQ [WS98]. According to Slater [Sla99], the PQ measures how users respond to the attributes of the system used rather than presence. However, we suggest that these questions do not have the aim to target presence, rather the aspects that Slater [Sla99] mentions. These questions could be used in further studies where 360° videos and VEs are compared or other studies where expectations and perceptions of affordances in VR are evaluated. Postural responses as used in the study by Freeman et al. [FAM*00] might also be an alternative for such studies but where the method is applied in order to measure expectations rather than presence.

In future studies similar to this, where realism is not the main goal, it is suggested to exclude questions that include the word "realistic" and perhaps even the word "real" in order to minimize the risk of confusion of what is actually measured. We need terms, or rather, words, that target the feeling of *being there*. The word needed could be "presence", but not as *the concept* of presence, but as a *noun*, meaning the feeling of being in the environment. When asking questions about presence, perhaps no underlying framework or intricate definitions with multiple dimensions should be used due to that participants do not always share this framework [MAT00].

There were no significant differences between the results

^{© 2017} The Author(s) Eurographics Proceedings © 2017 The Eurographics Association.

on General presence in the IPQs in this study. The item was as follows: "In the virtual environment/360°-video I had a feeling of "being there"." (in the original IPQ the item is taken from Slater & Usoh [SU94]). A similar question was asked in the interviews: "During the experiment, did you think that you actually were in any of the environments?" In the interviews, a majority of the participants clearly stated that this feeling was stronger in the 360° video than the VE. However, taking a closer look at these participants' answers on the item in the IPQ that targets General presence, some interesting results were found. One of the 6 participants scored the same for both environments and another actually reported a higher score for the VE even though having mentioned feeling there to a greater extent in the 360° video in the interview. The result might be due to poor choice of words in the interview question. By asking the participants if they "thought that they actually were there" it could be possible that people are reluctant to report a higher score on the scale since it might sound as if hypnosis or psychosis was involved. Lombard & Ditton [LD97] discuss this with regard to the illusion of nonmediation and claim that this illusion does not include confusion of what is real or not or psychological defects. However, this is something that probably is obvious to most researchers but not to the participants of a study. For future research where presence is measured we suggest to try the simple question: "Did you have a feeling of 'being there'?"

Treating presence as a noun that indicates the feeling of being in a place, rather than a term consisting of multiple aspects, we could finally complete the one mission many researchers in the area of VR are dealing with, namely trying to find an appropriate definition of presence. However, this does not mean that presence as the noun "feeling of being there" is the one and only aspect that needs to be evaluated in VR. We suggest that researchers should carefully choose aspects that are necessary to evaluate in order to achieve their goal. In video games, perhaps *involvement* is the main goal that should be strived towards. In a project where the aim is to simulate an already existing environment, such as the Eiffel tower or a botanical garden, the main goal may be that it looks realistic and real, that it looks and feels as the actual place. In a science fiction 360° movie, the main goal can perhaps be to make it *believable*. In a VR experience that includes meditation or mindfulness, presence could be the main goal.

To summarize, we suggest that presence is not always the main goal of a VR experience and VR research should not by default aim towards it. However, presence as the noun "feeling of being there" may in some projects be an aspect to strive towards.

5.3. Measuring presence

It is here argued that presence should not automatically be seen as the true goal for VR. However, some projects may include presence as a goal and a measurement that could be applied across media is thus needed in order to investigate if, how and why presence should be evaluated. By treating presence as a noun, meaning the feeling of being in a place, the questions could be easier to answer and it might be easier applying the same instrument across media.

Depending on the techniques used in a VR system, questions regarding the different aspects could be examined in order to find out what needs to be improved in order to achieve the desired goal, for instance presence. Although previous studies may have evaluated how these aspects affect presence, they may not give appropriate answers since the method used in order to investigate presence may include multiple dimensions that do not align with the definition of presence used for the proposed study. For instance, Schuemie et al. [SAvdM^{*}05] concluded that a more natural locomotion technique leads to more presence. In their study, presence is measured using the IPQ which means that the categories General presence, Spatial presence, Involvement, and Experienced realism have been included to measure presence. However, the result only refers to presence as a whole, including all aspects mentioned above which means that we do not know how the different categories influenced the result. Results from the current paper indicate that the different categories in the IPQ may benefit from being treated as separate aspects rather than combined into a concept of presence.

Applying pictorial scales [WSPW15] might be a useful way of measuring presence since no words are used that could be confusing for the participants or difficult translating. The scale measured attention allocation, spatial situation model, self-location, possible actions, cognitive involvement, and suspension of disbelief. The scale is referred to as a presence scale, indicating that all categories together can create the level of presence. A confusing aspect is that spatial presence and presence are treated as the same thing in the study since they mention that presence can be referred to as spatial presence and that spatial presence often is referred to as "being there", which also is a common definition of presence [IFR00, WSPW15]. However in the study by Schuemie et al. [SAvdM^{*}05], spatial presence is seen as a sub-category of presence. Weibel et al. [WSPW15] thus used other definitions, determinants and categories for measuring presence than the ones used in the study by Schuemie et al. [SAvdM^{*}05], however, both studies claimed having measured presence, or more precisely, Weibel et al. [WSPW15] claim having measured spatial presence which they refer to as the same as presence. A pictorial presence scale such as the *Pictorial Presence SAM* might be a good alternative for measuring different aspects of a VE. However, for future work, we suggest to choose the categories of the scale relevant for the study and not treat the whole scale as a presence-indicator since all factors such as "possible actions" might not suit the system or goal used for all studies. For example, possible actions may not be a determinant if we aim at achieving presence in a 360° video that only includes the action of looking around.

The study by Dalvandi et al. [DRC11] included 360°

videos, and items from different categories of the IPQ were applied in order to measure presence. Ramalho & Chambel [RC13] also conducted a study where presence was measured using a system that included 360° videos where they used an adapted version of the PQ. Since the IPQ and the PQ include different items, the two studies have used different frameworks and approaches to presence, making it difficult knowing if the same aspect actually has been measured. As outlined in this report, there are various ways of measuring presence, often by evaluating multiple aspects of the concept but where the outcome represents all aspects as a whole, as the concept presence. Comparing results of presence where different dimensions, definitions or sub-categories of presence are used could thus be misleading and we propose that studies thus cannot fully rely on previous research of presence when concluding what is needed for achieving presence in the VE to be produced.

6. Conclusions and future work

This study had the main purpose of comparing a low-budget computer-generated VE to a low-budget 360° video viewed in VR mode. The most fruitful results were obtained from the interviews. As the project proceeded, the notion of presence turned out to be an interesting but also confusing aspect and we thus decided to investigate this deeper, concluding that the current conceptualisations and measurements of presence are problematic. Results show that the low-budget environments used in this study induced discomfort for several participants. This suggests that low-budget alternatives for VEs and 360° videos might be useful for projects where the aim is to induce fear, such as virtual reality exposure therapy.

Even though the majority of the participants mentioned that the environments looked similar, there are aspects that differentiate them. A study similar to this where more resources are applied to the creation of the environments making the environments more alike would minimize the risk of other aspects than the technique used to have an impact. However, in this study, the aim was to compare the two environments used, rather than comparing the techniques in general. The interviews highlight that discomfort can appear for different reasons and it is thus important asking more detailed questions if the goal is to target a specific aspect of discomfort. In such studies it may also be beneficial to perform pre-tests and exclude people who are more likely to experience motion sickness in order to eliminate variables that are not in focus. Since the environments in this study did not include any possible actions other than look around, there are various ways studies similar to this could be conducted. For instance, a 360° video that includes automatic movement could be compared to a VE including locomotion for the purpose of provoking fear or achieving realism.

Both the IPQ and the interviews also point out confusion around the words reality, real and realistic. We suggest choosing the word "real" in contexts where the overall feeling of realism in an environment is measured. If the aim for a

© 2017 The Author(s)

VE is not to reproduce reality, rather to reproduce realities that could exist, we propose that the word "believable" is used. This study has given clues that some people perceive more possible actions in a VE than in a 360° video. In future studies, questions around this topic could preferably be included in order to evaluate how people interact and adopt VR.

One of the most noteworthy conclusions drawn from this paper is the notion of not treating presence as a concept including complex underlying dimensions, and start treating it as the noun, the feeling of being there. In this way, an agreed definition could be set and it could be easier to measure presence across media which could be as simple as including following question: "Did you have a feeling of 'being there'?". We suggest that presence should not automatically be a goal for VR. However, if presence is the main goal of a VR project, it needs to be evaluated considering the techniques and goals of that specific study. Results from other studies measuring presence might not be relevant since presence may have been considered a concept including multiple aspects and it could be unclear what factors contributed to the results. We suggest that all factors and sub-scales used in current measurements (e.g. presence questionnaires and pictorial scales) should be treated separately in order to clearly state what has been measured, and when mentioning presence, it is only referred to the noun of being there without any underlying framework.

Acknowledgements

The author would like to thank Kai-Mikael Jää-Aro who was the supervisor for the project and provided valuable guidance and advice throughout the process.

References

- [AJGMRG11] ALSINA-JURNET I., GUTIÉRREZ-MALDONADO J., RANGEL-GÓMEZ M.: The role of presence in the level of anxiety experienced in clinical virtual environments. *Computers in Human Behavior* 27, 1 (2011), 504–512. 1
- [BBA*04] BAÑOS R. M., BOTELLA C., ALCAÑIZ M., LIAÑO V., GUERRERO B., REY B.: Immersion and emotion: Their impact on the sense of presence. *CyberPsychology & Behavior* 7, 6 (2004), 734–741. 1
- [BC06] BRAUN V., CLARKE V.: Using thematic analysis in psychology. Qualitative Research in Psychology 3, 2 (2006), 77– 101. 3
- [CSSS06] COELHO C. M., SANTOS J. A., SILVÉRIO J., SILVA C. F.: Virtual reality and acrophobia: One-year follow-up and case study. *CyberPsychology & Behavior 9*, 3 (2006), 336–341.
- [DRC11] DALVANDI A., RIECKE B. E., CALVERT T.: Panoramic video techniques for improving presence in virtual environments. In Proceedings of the 17th Eurographics conference on Virtual Environments & Third Joint Virtual Reality (2011), pp. 103–110. 1, 2, 3, 6
- [FAM*00] FREEMAN J., AVONS S. E., MEDDIS R., PEARSON D. E., IJSSELSTEIJN W.: Using behavioral realism to estimate presence: A study of the utility of postural responses to motion stimuli. *Presence: Teleoperators and Virtual Environments 9*, 2 (2000), 149–164. 1, 2, 5

Eurographics Proceedings © 2017 The Eurographics Association.

- [GT07] GREGG L., TARRIER N.: Virtual reality in mental health. Social Psychiatry and Psychiatric Epidemiology 42, 5 (2007), 343–354. 1
- [IFR00] IJSSELSTEIJN W. A., FREEMAN J., RIDDER H. D.: Presence: Concept, determinants and measurement. In Human Vision and Electronic Imaging V (2000), vol. 3959 of Proceedings of SPIE Conference. 1, 2, 6
- [Igr16] igroup presence questionnaire (IPQ), 2016. 3, 5
- [Ins03] INSKO B.: Measuring presence: Subjective, behavioral and physiological methods. In *Being There: Concepts, effects* and measurements of presence in synthetic environments, Riva G., Davide F., IJsselsteijn W. A., (Eds.). IOS Press, Amsterdam, 2003, ch. 7, pp. 109–119. 2
- [JP10] JUAN M. C., PÉREZ D.: Using augmented and virtual reality for the development of acrophobic scenarios. comparison of the levels of presence and anxiety. *Computers & Graphics* 34, 6 (2010), 756–766. 2, 3
- [KDSCH12] KWANGUK K., DAEYOUNG R., SUN K. I., CHAN-HYUNG K.: Provoked arrangement symptoms in obsessive-compulsive disorder using a virtual environment: A preliminary report. Computers in Biology and Medicine 42, 4 (2012), 422– 427. 2
- [LD97] LOMBARD M., DITTON T.: At the heart of it all: The concept of presence. Journal of Computer-Mediated Communication 3, 2 (1997). 1, 6
- [LDAR*02] LIN J. J. W., DUH H. B. L., ABI-RACHED H., PARKER D. E., FURNESS III T. A.: Effects of field of view on presence, enjoyment, memory, and simulator sickness in a virtual environment. In *Proceedings of the IEEE Virtual Reality Conference 2002* (2002), pp. 164–171. 2, 3
- [MAT00] MURRAY C. D., ARNOLD P., THORNTON B.: Presence accompanying induced hearing loss: Implications for immersive virtual environments. *Presence: Teleoperators and Virtual En*vironments 9, 2 (2000), 137–148. 2, 5
- [MFGR12] MARINI D., FOLGIERI R., GADIA D., RIZZI A.: Virtual reality as a communication process. Virtual Reality 16, 3 (2012), 233–241. 4
- [PSR13] PILLAI J. S., SCHMIDT C., RICHIR S.: Achieving presence through evoked reality. *Frontiers in Psychology* 4 (2013), 86. 5
- [RC13] RAMALHO J., CHAMBEL T.: Immersive 360° mobile video with an emotional perspective. In Proceedings of the 2013 ACM international workshop on immersive media experiences (2013), pp. 35–40. 1, 3, 7
- [SAvdM*05] SCHUEMIE M. J., ABEL B., VAN DER MAST C. A. P. G., KRIJN M., EMMELKAMP P. M. G.: The effect of locomotion technique on presence, fear and usability in a virtual environment. In *Euromedia* (Toulouse, Apr. 2005), pp. 129– 135. 6
- [SFR01] SCHUBERT T., FRIEDMANN F., REGENBRECHT H.: The experience of presence: Factor analytic insights. *Presence: Tele*operators and Virtual Environments 10, 3 (2001), 266–281. 2
- [Sla99] SLATER M.: Measuring presence: A response to the Witmer and Singer presence questionnaire. *Presence: Teleoperators* and Virtual Environments 8, 5 (1999), 560–565. 5
- [SU94] SLATER M., USOH M.: Representations systems, perceptual position, and presence in immersive virtual environments. *Presence: Teleoperators and Virtual Environments* 2, 3 (1994), 221–233. 6
- [SUS94] SLATER M., USOH M., STEED A.: Depth of presence in virtual environments. Presence: Teleoperators and Virtual Environments 3, 2 (1994), 130–144. 3

- [SUS95] SLATER M., USOH M., STEED A.: Taking steps: The influence of a walking technique on presence in virtual reality. ACM Transactions on Computer-Human Interaction 2, 3 (1995), 201–219. 1
- [TMTC03] TURNER P., MCGREGOR I., TURNER S., CARROLL F.: Evaluating soundscapes as a means of creating a sense of place. In Proceedings of the 2003 International Conference on Auditory Display (2003). 1, 2, 3
- [WBBL*15] WRZESIEN M., BOTELLA C., BRETÓN-LÓPEZ J., DEL RÍO GONZÁLEZ E., BURKHARDT J.-M., ALCAÑIZ M., PÉREZ-ARA M. A.: Treating small animal phobias using a projective-augmented reality system: A single-case study. Computers in Human Behavior 49 (2015), 343–353. 2
- [WHB*07] WIRTH W., HARTMANN T., BÖCKING S., VORDERER P., KLIMMT C., SCHRAMM H., SAARI T., LAARNI J., RAVAJA N., GOUVEIA F. R., BIOCCA F., SACAU A., JÄNCKE L., BAUM-GARTNER T., JÄNCKE P.: A process model of the formation of spatial presence experiences. *Media Psychology 9*, 3 (2007), 493–525. 1, 2
- [WS98] WITMER B. G., SINGER M. J.: Measuring presence in virtual environments: A presence questionnaire. *Presence: Teleoperators and Virtual Environments* 7, 3 (1998), 225–240. 2, 5
- [WSPW15] WEIBEL D., SCHMUTZ J., PAHUD O., WISSMATH B.: Measuring spatial presence: Introducing and validating the pictorial presence SAM. Presence: Teleoperators and Virtual Environments 24, 1 (2015), 44–61. 2, 6

M. Krone¹, K. Schatz¹, N. Hieronymus¹, C. Müller¹, M. Becher¹, T. Barthelmes¹, A. Cooper³, S. Currle¹, P. Gralka¹, M. Hlawatsch^{1,2}, L. Pietrzyk², T. Rau¹, G. Reina¹, R. Trefft¹, T. Ertl^{1,2}

¹Visualization Research Center (VISUS), University of Stuttgart, Germany ²SimTech Cluster of Excellence, University of Stuttgart, Germany ³Institute for Theoretical Chemistry, University of Stuttgart, Germany

Abstract

In this paper, we present the design considerations of a scientific exhibition we recently realized. The exhibition presented the work of two large research projects related to computer simulations, which include scientific visualization as an essential part of the involved research. Consequently, visualization was also of central importance for our exhibition. It was not only used to illustrate the complex simulation data to convey information about the results from the application domains, but we also wanted to teach visitors about visualization itself. Therefore, explaining the purpose and the challenges of visualization research was a significant part of the exhibition. We describe how we developed an engaging experience of a highly theoretic topic using the same visualization tools we developed for the application scientists and how we integrated the venue into our design. Finally, we discuss our insights from the project as well as visitor feedback.

Categories and Subject Descriptors (according to ACM CCS): K.3.2 [Computers and Education]: Computer and Information Science Education—Literacy I.3.8 [Computer Graphics]: Applications—Public Dissemination

1. Introduction

Scientific research, especially basic research conducted at universities, is often funded by government agencies, such as the German Research Foundation (DFG) in Germany. However, the results of such basic research are typically only published in scientific journals not targeting the general public. We see two potential reasons for this: On the one hand, scientific findings might be too complex to be explained to laymen. On the other hand, scientist might be interested mainly in informing their peers about new results, since this is more important for their individual careers than public dissemination. A promising strategy to make a broader audience aware of research and its relevance to society are science exhibitions in a public space. Such an exhibition should convey scientific approaches and basic research findings in an understandable and engaging manner. This way, it can additionally show the visitors that a publicly funded research project is not throwing the taxpayers' money out of the window, but rather produces significant results that might already influence everyday life, or at least will some day.

We report about the science exhibition "Im digitalen Labor – Durch Computersimulationen die Welt entdecken"

("In the Digital Laboratory - Understanding the world through computer simulations"), which is part of the public relations activities of the two research projects Cluster of Excellence Simulation Technology (SimTech) and Collaborative Research Center 716 (SFB 716) of the University of Stuttgart. We want to illustrate the design considerations behind our exhibition and explain the difficulties involved in a public presentation of basic research to non-experts. Since both projects make heavy use of computer simulations, visualization of the results is a central part of their research agenda. The scientists develop new visualization techniques not only for use by their colleagues in the application domain, but also for public dissemination. Therefore, our paper is not limited to showing how results from an application domain can be conveyed by visualization, but we also reports how we explain visualization research to the visitors.

While science museums often convey basic scientific knowledge in their exhibitions, our exhibition posed the above-mentioned design challenge to present recent research activities to the public. Psychological evaluations of scientific exhibitions are rather common, but their design is not really covered by scientific literature. Only few works address the difficulties of this phase of the exhibitions. Allen and Gutwill [AG09], for example, present the concept for a set of game-like exhibits that are designed to deepen the scientific inquiry skills of the audience while maintaining their interest at all times. Roche et al. [RCB16] and Bell [Bel00] examined the influence of science shows that were given during exhibitions like the one presented here. Maintaining the interest of the audience was regarded as easy, but at the cost of scientific profoundness. The design of single exhibits with certain intents was, for example, covered by Nim et al. [NWZ*16], who presented a virtual reality station communicating the dangers of coral bleaching in the Great Barrier Reef, by Hinrichs et al. [HSC08], who investigated the use of touch-interaction and information visualization in a museum exhibition, or by the public presentation framework for molecular data by Nickels et al. [NSM*13].

2. Exhibition Concept

Our exhibition tries to convey a holistic view on the state of the art in computer simulations. Such simulations are, besides experiments and theories, an essential part of today's research and development processes. Scientific progress in a wide area of applications from life sciences, engineering, physics, chemistry to materials sciences is hardly imaginable without them. They are used to design models, to verify processes in nature and technology, or to explore possible scenarios of the future. Thus, they have become highly relevant to our modern society as an essential part in the development processes of many everyday products like microchips, but also as a tool to find the right therapies in medicine. This ubiquity of computer simulations in natural sciences stands in contrast to public knowledge about them, which was the reason for us to design a science exhibition explaining the imperative nature of computer simulations to the public.

Simulation technologies are an important research area at the University of Stuttgart and core interests of the two projects involved in the exhibition: The focus of SFB 716 is the development of dynamic simulation methods for systems with large numbers of particles. Scientists use numerical simulations to answer questions about material properties, processes in biochemistry and chemical engineering. The amount and complexity of the data involved mandates the development of efficient algorithms, coarsening and acceleration mechanisms, and special visualization techniques. Simulation sciences as a whole are the research are of the interdisciplinary SimTech research association. More than 200 scientists from virtually all faculties of the university are working together to reach a common goal: developing simulation technology into an integrative systems science. Both SFB 716 and SimTech could therefore contribute their expertise, technical infrastructure and staff to the exhibition.

We chose Carl-Zeiss-Planetarium Stuttgart as the location for the exhibition, for a planetarium being a place of non-formal education and knowledge transfer. Bridging sci-



Figure 1: Overview of the exhibition space. In the foreground, one of the four pedestrals can be seen. In the background, the room with the interactive stations is visible.

ence and entertainment, it offers the perfect environment for our purposes. The exhibition was shown for three months in the public area of the planetarium. It explains the world of computer simulations, their development and how they work. A special focus is on the presentation of visualizations from various research areas, ranging from still images over movies to interactive software applications. The target audience is the general public, not necessarily with an academic background, with a focus on people with a general interest in science, high-school and university students.

The exhibition is structured into several stations. Fig. 1 shows an overview of the exhibitions space. One of them shows the history of simulations beginning with Navier and Stokes and ending with today's tasks and challenges. Each of the others is dealing with an individual aspect of simulations. Four of the exhibits are pedestals combined with an information panel. We use their front and rear side:

Pedestal I: Idea of Simulation

- Front: "What is a Simulation?" reasons for simulations, like cost or risks of experiments *Interaction:* six rotatable panels, *exhibit:* natural gas injector in simulation and prototype
- Back: "This is how it works" simulation process *Interaction:* tablet PC with simulation of human gait

Pedestal II: Background

- Front: "From Reality to Model" first steps, formulation of a scientific question *Interaction:* turntable explaining different approaches to
- formulate a scientific issueBack: "Reality in Equations": particle model/mesh model
- Interaction: hands-on atomic model and mesh model

Pedestal III: Technology

• Front: "Simulation Equipment" – simulation on personal computers and high performance compute clusters *Exhibit:* decommissioned blade of *Hermit* supercomputer from the Stuttgart HPC center (HLRS)

• Back: "Smart Computing" – efficient computing with streaming, parallelization, coarsening etc. *Interaction:* puzzle game with assignments

Pedestal IV: Visualization

- Front: "What is Visualization?" reasons for visualization of scientific data *Interaction:* find the match – cards with visualizations and questions have to be assigned to the correct answer
- Back: "From Data to Images" visualization pipeline Interaction: tablet PC with HTML slider explaining the steps for the visualization of a virus

The last station at the center of the exhibition is a small room having information panels at its outside and exhibits of interactive visualization inside. While the previous exhibits provide background information, this one shows what is at the end of the simulation process: the results of simulation research and specific applications in practical use in science and industry, illustrated by aesthetically pleasing scientific images and visualizations from various research areas.

Central Cubicle

- Lockable room within the exhibition hall $(3 \times 3 \text{ m})$
- Exterior walls show vivid examples: text and visualizations of applications, examples, results
- The four exterior walls offer the possibility to group images into four thematic areas: material design; machine engineering, process engineering & technology; biotechnology & medicine; nature & environment
- Interior: two screens showing interactive visualizations steered by motion control systems (*Microsoft Kinect, Leap Motion*) via the visitors' hands

For creating the exhibition's concept, we followed three principles: to show what a simulation is and how it works; to make a complex and abstract scientific topic comprehensible for everyone; and to present research from the perspective of the scientists themselves. The practical implementation of the concept posed several important challenges:

- Every station should deal with an individual topic and stand for itself, i.e. it should be usable without the context of the whole exhibition.
- The stations should work without guided exhibition tours.
- The stations' interactions should literally be touchable and invite to playfully discover the topic in order to create a better learning effect.
- The stations and in particular the analog and digital interactions and exhibits should be theft-proof and indestructible – at least by involuntary or casual vandalism.

3. Visualization and Interaction Concepts

Explaining the results of simulation research to a broader non-academic audience poses several challenges. The main task is to enable even high-school students to understand key concepts of the ongoing research. This can only be achieved



Figure 2: The interactive visualization exhibit. The visualization on the left screen is steered using a Microsoft Kinect, the right one via Leap Motion.

by the use of visualization techniques, since the real-world counterparts of the visualized phenomena may be to big or too small to be examined and most of the resulting data may be too abstract to be understandable right away. Thus, nearly all exhibition pedestals offer some kind of visualization. In fact, the cubicle featuring interactive visualizations of simulation results is the centerpiece of the exhibition (Fig. 2).

A second challenge is to maintain the interest of the audience during the whole visit. Since purely static exhibits may be considered boring (especially by a younger audience), we decided to add interactive and tactile exhibits. This includes the gesture-controlled interactive visualizations shown in Fig. 2 as well as 3D-prints of previously visualized objects.

3.1. Interactive Stations

The above-mentioned cubicle, which is the centerpiece of the exhibition, is a small room of approximately 3×3 meters that features two interactive stations where visitors can interact with visualizations. Two high-resolution 32'' LCD displays are mounted on the rear wall of the room (see Fig. 2). Each display is driven by a dedicated visualization workstation, which is hidden from the visitors inside a double wall of the room. The space within the double wall (approximately 40 cm) is open to the top and accessible via a small lockable hatch to allow for sufficient air flow and maintenance.

Our visualization framework MegaMol [GKM*15] served as a basis for the interactive stations. MegaMol is the rapid prototyping tool for visualization of particle-based simulation data within SFB716. It offers all the interactive scientific visualization methods for molecular data like molecular surfaces, ball-and-stick models, or particle trajectories, which have been developed by the visualization researchers of the project. That is, the visual representations developed in the collaborative research project were already available for use in the exhibition context.

Selection of the Examples: Each station allows the visitors to explore one simulation data set. The choice of the data to



Figure 3: Screenshot of the laser ablation simulation used in the interactive station that is steered via Microsoft Kinect.

present was driven by the idea that it should show a sufficiently interesting scientific phenomenon and that the visual representation should also look interesting.

The first data set is a simulation of a laser ablation, a process used in many industrial applications. Here, a laser beam hits a block of aluminium, causing a small portion of the metal at the surface to heat up and explode outwards, thereby creating a small crater. In the simulation, the aluminum block consists of approximately 560 k particles. The user can switch between a visualization of the raw particles and an abstract visualization where only the trajectories of larger particle clusters are shown [FSG^{*}11] (see Fig. 3).

The second data set is a simulation of a lipase protein, which is part of the fat metabolism. In the simulation, the lipase sits atop a layer of fat molecules, while one fat molecule moves towards the active site of the protein where the fat would be split (see Fig. 4). To show the complexity of the protein and to allow the user to see the fat moving into a tunnel within the protein, we chose a semi-transparent molecular surface visualization [KSES12]. The complex structure of the lipase is shown using the Cartoon representation within the surface [KBE08]. Both protein representations are rendered in a medium blue color, to show that both geometries visualize the same molecule. Fat molecules in the layer are colored light yellow and displayed as Stick representation. Only the specific fat molecule that moves into the protein is rendered in red to visually stick out.

User Interaction: One major challenge when designing the interactive stations was to come up with a interaction concept that is both easy to use and engaging for the visitors. Users should be able to grasp the interaction possibilities either intuitively or with very little explanation provided on a small poster next to the interactive station. Another consideration was that the users should not be able to exit the visualization by accident or on purpose. One obvious choice would be to use a touch screen as a replacement for the typical mouse interaction, such as the touch tables used by Ynnerman et al. [YRA*16] and Agus et al. [AMB*17] in museums. However, we saw several problems with this concept. First, large touch screens are still quite expensive compared



Figure 4: Screenshot of the lipase simulation used in the interactive station that is steered via Leap Motion.

to standard non-touch displays. Second, a user interacting with the visualization would occlude parts of the screen from other visitors since he/she would have to stand close in order to touch the screen. Third, extra measures have to be taken to prevent users from exiting the application, since the touch screen would effectively act as a mouse. Therefore, we decided to implement a gesture-based system that uses either Microsoft Kinect or Leap Motion as input device. While the Kinect can be placed on top of the display using a commercial mounting kit, the Leap requires a table that places the device about 30 cm underneath the hand of a standing adult person. According to the DIN 33402 standard, the average elbow height in Germany is 106.75 cm, therefore, a table height of 75 cm was chosen. We fabricated a special mount for the Leap using 3D-printing that was screwed to the table to prevent theft. The cables of both devices are also mostly hidden from visitors to prevent casual vandalism or accidental unplugging. Both devices are provided with software development kits that allow for a relatively simple usage. Due to the modular concept of MegaMol, adding support for these devices into the existing interaction concept required no changes of the software architecture.

Camera Adjustment via Gestures: The basic interaction with the visualization is the manipulation of the camera. Users should be able to turn the data around and to zoom in an out. Modifying the camera uses the same gestures on both devices: A simple "grab" gesture (i.e., making a fist) activates the camera adjustment. If the fist is moved left and right or up and down, the camera orbits the data sets (i.e., the data set is turned around). Moving the fist towards or away from the screen zooms in and out. To exit the camera adjustment mode, the user simply has to open the fist again.

Graphical User Interface: Since the desired interactions also include parameter changes, a simple graphical user interface (GUI) that can be steered via gestures was also needed. The standard GUI of MegaMol is too complex for this application case and warrants a mouse and keyboard.

The interaction with the GUI mainly involves selecting buttons. Due to the different accuracies of the two devices,



Figure 5: 3D printed molecular surface of two proteins.

we chose to use slightly different gestures here. For the Leap Motion, the user has to extend his/her index finger and point at the screen. While the index finger points at the screen, a ring-shaped cursor is shown on the screen at the position the user points at. Since Kinect was not able to reliably detect the extended index finger, we decided to use an extended flat hand to activate the cursor mode. If the user moves the cursor over a button, the cursor ring starts to fill. Once the whole ring is full, the button is activated. We decided to use this technique since it is routinely used by Kinect games, therefore, we reckoned that it would be familiar to users knowing this gaming console. The animation of the ring that is filled up also makes the method intuitive and easy to explain, as it provides clear graphical feedback. As mentioned above, the GUI itself consists of buttons that can be selected by the user. Buttons can have a text label and an icon. As observable in Figures 3 and 4, we also added a short text that explains the visualized simulation.

Since users are typically much less accurate when using mid-air gestures than when using a mouse [VMM14], the buttons are relatively large in relation to the screen size to make it easy to keep the cursor on them. Additional buttons known from AV devices (play, pause, rewind), which feature no text but only the well-known media control symbols (see Fig. 3) can be used to control the playback of the time-dependent simulation data.

3.2. Tangible Visualization

On-screen visualization can only approximate the appearance of objects as they would look in the real world. It can be beneficial to complement the visual representations with tangible ones, which has become possible and affordable through 3D printing. This enables visitors to get a more complete impression. We, therefore, included a 3D printed object in the pedestal targeting molecular surface visualization. While the visualization technique is explained in textual form, the actual result is displayed as 3D printed model.

Creating such models requires several steps: First, the surface of the model has to be computed. This is done using our visualization framework MegaMol, which is able to compute the Solvent Excluded Surface [Ric77] via the MSMS algorithm [SOS96]. The result of this computation is a triangle mesh that can be used as input for the 3D printing.

In 3D printing, the choice of the material and printing technology is crucial for a suitable result. In our case, the material had to be stable enough to survive several months of skin-contact and potentially rough handling, the printing technology has to be able to cope with the complex structures of a molecule as well as a large number of overhangs. Depending on the printing technique, these overhangs may need support structures to be printed correctly. We opted for laser-sintering, since it does not need support structures. As material, we chose a polyamide that is sufficiently robust. To reduce production costs, the models have been printed hollow. The polyamide only allows for a single color. While multi-colored 3D printing is possible in principle, the resulting objects would have been too fragile for our purpose. The 3D printed molecules shown in Fig. 5 are fixed to the visualization pedestal using thin wire ropes to prevent theft.

3.3. Explaining Scientific Visualization to Laymen

The two interactive stations explained in Section 3.1 only use visualization as a vehicle to show simulation data to the user. Besides using visualization to convey simulation results, one goal of the exhibition was to explain the process of visualization and the research involved in this field of research to the visitors. However, most scientific visualization algorithms are too complex to be explained to laymen in the context of a self-guided exhibition, with visitors only getting information from reading the text provided on the pedestals. We therefore decided to explain the process and challenges involved in scientific visualization research in two simple ways. First, the pedestal dedicated to visualization (IV, see Section 2) briefly summarizes the visualization pipeline using the example of molecular surface visualization. The textual explanations are backed by graphical depictions of the individual stages (raw input data, filtering, mapping, renderable representation). Second, we designed an exhibit showing the complexity of scientific visualization and computer graphics using the example of lighting calculations.

Lighting is not only a very important factor in data visualization, it is also easy to understand for a general audience without prior knowledge, since it models a natural phenomenon everybody knows. Furthermore, the individual steps involved in the lighting calculation are in general understandable without knowing the calculations involved. To illustrate this, we used a 12.3" tablet PC (Microsoft Surface Pro 4). We created an HTML application which allows the user to swipe or navigate using buttons through the steps of the computation. For each step, a short textual explanation is displayed in conjunction with an image showing the effect. Our example shows a virus capsid consisting of approximately 200,000 atoms that form a spherical shape.

M. Krone et al. / From Visualization Research to Public Presentation – Design and Realization of a Scientific Exhibition



Figure 6: Screenshot of the HTML page explaining lighting calculations in scientific visualization. The leftmost image shows the full slide layout with an image of the final visualization result and a short description. The rest of the images show example images of the different stages (from left to right: without lighting, local lighting, shadows, ambient occlusion).

Each atom is depicted as a sphere. The first image shows the virus without any lighting, that is, everything appears flat. In the second image, local Blinn-Phong shading is employed, showing the atoms' spheres as round objects. Images four and five show the effects of shadow mapping [KRZ*17] and ambient occlusion [GKSE12], which help to convey the global spherical shape of the virus. In the last image, all effects are combined to show how each of them contributes to the final appearance. Fig. 6 shows the slide layout and example images. To prevent theft, the tablet PC is locked into a commercial metal mount that is fixed to the pedestal. Furthermore, we disabled all OS-level touch gestures in Windows 10 that would allow a user to interact with the operating system and locked the browser displaying the HTML slides to kiosk mode.

4. Fulldome Movie

Given the planetarium as our location, we decided to use all of the assets available there, namely the fulldome video projection system. Therefore, we produced a ten-minute documentary movie to be shown before the normal program of the planetarium during the exhibition, advertising the exhibition to all visitors of the planetarium. The movie features several of the visualizations for simulations developed within the two research projects. It explains the necessity and usefulness of simulations and their subsequent visualization to the audience. The visualization sequences include results from SFB 716 of laser ablation simulations, protein interactions, DNA transport, and studies of the cracking behavior of aluminum (see Fig. 9). Additionally, we included a visualization of a simulation of underground carbon dioxide storage that covers a period of over one hundred years. All renderings of particle simulations are directly taken from Mega-Mol, our visualization framework for large particle data sets, which is also used by the domain scientists. We also produced a significant amount of cutscenes between the actual visualization sequences, which have been produced using Maya and Mental Ray (see Fig. 7).

The key difference between this movie and regular videos is the non-planar projection surface. Planetariums handle this spherical projection by means of a special image format called *dome master*. A dome master is a square image embedding a fish eye projection of the dome's content (Fig. 8). Videos in this format can normally be shown on all fulldome systems, possibly after some post-processing to adapt to the hardware characteristics of the specific planetarium.

There are basically two ways of producing such dome masters from computer-generated imagery: When using ray casting, one could generate rays mimicking the characteristics of a fish eye lens in the first place. Or one could render at least five sides of a sky box an reproject their content to the dome master – an approach which is compatible with any traditional interactive rendering approach. We opted for the second way, because it allows us to use any visualization technique already available in MegaMol, regardless of the rendering technique. We were able to implement this with only minimal changes to our existing code base: Besides adding the option to render the sides of the sky box with fixed 1:1 aspect ratio and aperture angle of 90° , we had to implement the means for animating the camera. We developed a key frame-based camera path editor to create smooth camera movements for the movie. The camera positions between the key frames, as well as other camera-parameters, are interpolated using Catmull-Rom splines, which ensure that the camera actually passes all key frame positions. We also interpolate the simulation data to produce smooth material in 30 fps. For scientists, the low number of snapshots exported from the simulations is not a problem when interactively exploring their simulation data. However, creating a film directly from those snapshots can result in jumping particles if the positions are not interpolated. Therefore, we had to implement an interpolation of the particle positions that could also handle periodic boundary conditions usually found in particle simulations. These boundary conditions cause particles leaving the simulation domain on one side to re-enter it on the opposite one to conserve the energy.

The original idea was to perform the projection from the sky box to the dome master during post-production using commercial plugins available, e.g., for *Adobe After Effects*. However, we quickly realized that it is difficult to design meaningful sequences filling appropriate parts of the dome without an actual preview rendering. We thus developed an additional tool that allowed us not only to verify the dome master embedding by producing images like Fig. 8, but also to render actual preview videos. The latter allowed us to get a



Figure 7: Closing credits of the fulldome movie, photographed from the ground upwards.

feeling for how fast objects move on the dome's surface and to verify that the camera paths make sense in their entirety.

To guide the audience through the single visualizations and to keep them engaged between the sequences, we used traditionally animated sequences produced in Autodesk Maya, which can directly render dome masters by means of an appropriate plugin. To prevent distracting the viewer from the displayed information, scientific visualization tends to be much more mundane than the computer-generated imagery that we have become used to from cinema. Meeting the expectation of the audience for an animated movie while still matching the tone of our scientific visualizations was the biggest challenge for producing cutscene animations. To have a consistent theme throughout the movie, the cutscenes also feature a particle-based simulation, although a very simple, purely synthetic one. As in our scientific visualizations, particles are rendered as simple spheres but with more sophisticated surface shading. Scene lighting is composed of a single back light to give the scene a more dramatic atmosphere and to generate some contrast to the more evenly lit scientific visualizations. To improve the perception of scale and depth, the scene is located in a volumetric nebula and employs volumetric lighting (see Fig. 7).

To match the resolution of the fulldome system, we produced our movie using $6,122 \times 6,122$ px (6K) dome masters. For the MegaMol scenes, these have been reprojected from 4K sky box sides, except for the laser beam data set (fourth in Fig. 9). The high number of particles in this data set demanded supersampling to reduce aliasing artifacts, wherefore we rendered each side in 6K. We ended up having a 1.1 TB of raw imagery in total, visualization sequences and cutscenes combined. Rendering the latter took about nine days in total on a cluster of 20 nodes. The same cluster ran almost 16 days to render the sequences in MegaMol. Note that only a small fraction of this imagery found its way to the final film. The overwhelming majority of the rendering time was consumed for iterating over the sequences until the best camera path and visualization parameters had been found.



Figure 8: A dome master frame from the visualization of a laser ablation simulation. The parts from different sides of the sky box are subtly tinted to show the production process.

5. Summary and Conclusion

In the first month, over 3,500 visitors attended the exhibition and its associated events (vernissage, scientific lectures for the public). During this time, we collected visitor feedback using anonymous questionnaires that allowed visitors to rate and comment on different aspects of the exhibition. As expected, only a small fraction of the visitors provided feedback (25 completed questionnaires; 52 % female visitors, 36% male, 12% unknown; 80% over 13 years old, i.e., fitting the intended audience). Even though the number is quite low, this initial feedback already provides us with valuable insights. The explanations about the simulation and visualization techniques were overall regarded as good and comprehensible, while not being too simple. Especially the interactive exhibits that allowed visitors to explore scientific visualizations were consistently rated as very good and engaging. The dome movie also received very positive marks and comments, which is a very pleasing result, since it was intended to catch the interest of visitors that originally only planned to watch a planetarium show. Even younger visitors (<13 years) rated the exhibition as interesting and comprehensible, which we did not expect. We guess that these younger kids were accompanied by adults who guided them. One person actually commented that the exhibition was too small. Although this is only one individual sentiment, we infer from the overall positive feedback that our exhibition concept was engaging and did not overwhelm the visitors.

We of course also encountered some issues that could be improved in the future. Although we designed all exhibits to be as sturdy as possible, we encountered a few cases that we did not expect to fail. As mentioned in Section 3.2, we chose a polyamide material for the 3D printed molecules due to its robustness, however, since this material is dyed after printing, the color slightly wears off after some time. Using a colored material would circumvent this issue. Regarding the fulldome movie, the circular seating necessitates that the movie has no orientation – that is, no "up" and "down". While this is not critical for most planetary movies, it is a



Figure 9: Several scenes shown in the fulldome movie. From left to right: Laser ablation with the two moving lasers forming a cross on the surface. DNA strand tunneling through a nanopore. Crack propagation through aluminium. Laser ablation with one static laser. Lipase trying to detach fat from a larger structure. All pictures were rendered using MegaMol.

bit harder to create such a movie for particle simulations. Although we paid attention to this fact, at least one visitor comment was that this could be improved. For the interactive stations, the Leap Motion proved to be better suited than the Microsoft Kinect. Kinect is not only less accurate in our setting – which might be due to lighting conditions – but also has the issue that users need to stay away a certain distance from the device. Although we marked the optimal distance on the floor, visitors tend to stand too close. Another issue is that the Kinect sometimes switches to another person if the room is crowded. This could be fixed by implementing a more stable tracking.

In summary, we have presented our concept for a scientific exhibition for the general public that makes heavy use of scientific visualization. We have discussed our design choices and explained that we not only used visualizations to convey simulation research, but also presented visualization as an area of research to the visitors. As mentioned above, we are aware that the evaluation of the questionnaires has to be taken with a grain of salt due to the low number of completed questionnaires. However, we still think that the very positive feedback shows us that the design of our exhibition was overall successful.

Acknowledgments: This work was funded by German Research Foundation (DFG) as part of SFB 716 and SimTech Cluster of Excellence. We would like to thank the staff of Carl-Zeiss-Planetarium Stuttgart, especially the director, Uwe Lemmer, who gave us the opportunity to host the exhibition in their wonderful location, and Ubbo Grassmann who provided invaluable support for the creation of the dome movie. All photographs are courtesy of Max Kovalenko.

References

- [AG09] ALLEN S., GUTWILL J. P.: Creating a Program to Deepen Family Inquiry at Interactive Science Exhibits. *Curator: The Museum Journal* 52, 3 (2009), 289–306. 2
- [AMB*17] AGUS M., MARTON F., BETTIO F., HADWIGER M., GOBBETTI E.: Data-driven analysis of virtual 3d exploration of a large sculpture collection in real-world museum exhibitions. *ACM Journal on Computing and Cultural Heritage* (2017). 4
- [Bel00] BELL T.: A Low-cost High-impact Computer Science Show for Family Audiences. In Australasian Computer Science Conference (2000), pp. 10–16. 2

- [FSG*11] FREY S., SCHLÖMER T., GROTTEL S., DACHS-BACHER C., DEUSSEN O., ERTL T.: Loose Capacity-Constrained Representatives for the Qualitative Visual Analysis in Molecular Dynamics. In *IEEE PacificVis* (2011), pp. 51–58. 4
- [GKM*15] GROTTEL S., KRONE M., MÜLLER C., REINA G., ERTL T.: MegaMol – A Prototyping Framework for Particlebased Visualization. *IEEE Trans. Vis. Comput. Graphics* 21, 2 (Feb 2015), 201–214. 3
- [GKSE12] GROTTEL S., KRONE M., SCHARNOWSKI K., ERTL T.: Object-Space Ambient Occlusion for Molecular Dynamics. In *IEEE Pacific Visualization Symposium* (2012), pp. 209–216.
- [HSC08] HINRICHS U., SCHMIDT H., CARPENDALE S.: EMDialog: Bringing information visualization into the museum. *IEEE Trans. Vis. Comput. Graphics* 14, 6 (2008), 1181–1188. 2
- [KBE08] KRONE M., BIDMON K., ERTL T.: GPU-based Visualisation of Protein Secondary Structure. In *Theory and Practice* of Computer Graphics (2008), vol. 8, pp. 115–122. 4
- [KRZ*17] KRONE M., REINA G., ZAHN S., TREMEL T., BAH-NMÜLLER C., ERTL T.: Implicit sphere shadow maps. In *IEEE PacificVis - Visualization Notes* (2017), vol. 4. 6
- [KSES12] KRONE M., STONE J. E., ERTL T., SCHULTEN K.: Fast Visualization of Gaussian Density Surfaces for Molecular Dynamics and Particle System Trajectories. In *EuroVis Short Papers* (2012), vol. 1, pp. 67–71. 4
- [NSM*13] NICKELS S., STÖCKEL D., MUELLER S. C., LENHOF H.-P., HILDEBRANDT A., DEHOF A. K.: Presenta-BALL — a Powerful Package for Presentations and Lessons in Structural Biology. In *IEEE BioVis* (2013), pp. 33–40. 2
- [NWZ*16] NIM H. T., WANG M., ZHU Y., SOMMER B., SCHREIBER F., BOYD S. E., WANG S. J.: Communicating the Effect of Human Behaviour on the Great Barrier Reef via Mixed Reality Visualisation. In *BDVA* (2016), pp. 1–6. 2
- [RCB16] ROCHE J., CULLEN R. J., BALL S.-L.: The Educational Opportunity of a Modern Science Show. *International Journal of Science in Society* 8, 3 (2016). 2
- [Ric77] RICHARDS F. M.: Areas, Volumes, Packing, and Protein Structure. Annu. Rev. of Biophys. Bio. 6, 1 (1977), 151–176. 5
- [SOS96] SANNER M. F., OLSON A. J., SPEHNER J.-C.: Reduced Surface: An Efficient Way to Compute Molecular Surfaces. *Biopolymers* 38, 3 (1996), 305–320. 5
- [VMM14] VRELLIS I., MOUTSIOULIS A., MIKROPOULOS T. A.: Primary school students' attitude towards gesture based interaction: A comparison between microsoft kinect and mouse. In *IEEE ICALT* (2014), pp. 678–682. 5
- [YRA*16] YNNERMAN A., RYDELL T., ANTOINE D., HUGHES D., PERSSON A., LJUNG P.: Interactive visualization of 3D scanned mummies at public venues. *Communications of the ACM* 59, 12 (2016), 72–81. 4

Evaluating the Influence of Stereoscopy on Cluster Perception in Scatterplots

Christian van Onzenoodt, Julian Kreiser, Dominique Heer, Timo Ropinski¹

¹ Visual Computing Group Ulm University, Ulm, Germany {christian.van-onzenoodt, julian.kreiser, dominique.heer, timo.ropinski}@uni-ulm.de

Abstract

Unlike 2D scatterplots, which only visualize 2D data, 3D scatterplots have the advantage of showing an additional dimension of data. However, cluster analysis can be difficult for the viewer since it is challenging to perceive depth in 3D scatterplots. In addition, 3D scatterplots suffer from overdraw and require more time for perception than their 2D equivalents. As an approach to this issue, stereoscopic rendering of three-dimensional point-based scatterplots is evaluated through a user study. In detail, participants' ability to make precise judgements about the positions of clusters was explored. 2D scatterplots were compared to non-stereoscopic 3D and stereoscopic 3D scatterplots. The results showed that performance in perception decreased when confronted with 3D scatterplots in general, as opposed to 2D scatterplots. A tendency towards an improvement of perception showed when comparing stereoscopic 3D scatterplots to non-stereoscopic 3D scatterplots.

1. Introduction

Finding clusters in scatterplots is a common task in data visualization. Whether it is to analyze the relationship between two or more discrete variates, to observe the formation of many clusters, or to find trends and correlations, scatterplots are widely used and well-known diagrams to represent data. While lots of effort has been put into the investigation of 2D scatterplots, plots with three dimensions have not been investigated deeply. Even though there are several techniques to render high-dimensional data, the quality of perception still needs to be further investigated. It has been noted that the human visual system can reliably perceive clusters, trends, and correlations in sparse 2D scatterplots [LMvW10b, LMvW10a, G*13]. Even if the density increases, techniques exist to reduce clutter in scatterplots (see Section 2).

However, when dealing with 3D scatterplots, it is often considered difficult to make precise judgements about the features of clusters. Like printed pages, common computer monitors are only able to effectively display 2D images. When viewing a static 3D diagram on such monitors, it is hard to perceive the location of a single point, let alone the expansion of a cluster, due to the incapability of perceiving depth in a 2D representation. In computer graphics, various techniques have been developed to add depth perception to 2D images, such as occlusion, shading and defocus blur [Meh13]. However, most techniques are not practical when dealing with scatterplots. In the last few years, active stereoscopic displays were developed and made available on the consumer market. By rendering two separate images and presenting them to both eyes individually, a real 3D effect can be achieved. This technique is well-known as active stereoscopic rendering by using shutter glasses. Therefore, in this paper a user study is presented to compare viewers' ability to perceive clusters in three different classes of scatterplots: 2D scatterplots, 3D scatterplots, and stereoscopic 3D scatterplots. More precisely, a set of 2D and 3D scatterplots was created. In each diagram, one cluster was marked in a different color than all other clusters (see Figures 1 and 2). Then, user experiments were performed to compare viewer's ability to make precise judgements about the location of the marked cluster in 2D scatterplots, 3D scatterplots and stereoscopic 3D scatterplots.

Among many other techniques for visualizing multivariate data, scatterplots are widely used. Although invented in the first half of the nineteenth century, scatterplots are very common when it comes to data display. In fact, between 70 and 80% of graphs used in scientific publications are scatterplots [FD05]. They are especially useful when analyzing the relationship between two or more variables. Figure 3 shows a typical 2D scatterplot featuring three different classes of data (red, blue and green).

Scatterplots are not limited to show only 2D data; in principle, arbitrary high-dimensional data can be displayed when using computer systems. In 1983, D. Asimov invented the *grand tour*, a technique to project high-dimensional data orthogonally onto some 2D subspace. It is thereby possible to view higher-dimensional data as a sequence of carefully chosen 2D scatterplots [Asi83]. Due to technical progress in the field of computer graphics, 3D scatterplots have become a major field of interest in the last few years. While

^{© 2017} The Author(s) Eurographics Proceedings © 2017 The Eurographics Association.



Figure 1: Exemplary 2D scatterplot as shown to the participants. The plot shows three clusters; the orange (marked) cluster has its center at (x = 6, y = 3).



Figure 2: Exemplary 3D scatterplot as shown to the participants. The plot shows five clusters; the orange (marked) cluster has its center at (x = 4, y = 0, z = 1).

showing one more data dimension, it is often difficult to perceive depth in 3D scatterplots. Therefore, a number of methods were developed and evaluated to improve depth perception in 3D scatterplots, such as illuminated scatterplots [SW09]. On the one hand, it has been shown that most techniques do indeed improve depth perception. On the other hand, these methods usually cannot compensate the absence of a third dimension.

2. Related Work

Evaluating Scatterplots. Due to the prevalence of scatterplots in scientific applications, a rich literature exists. Poco et al. [PEP*11] conducted a user study to compare point-based 3D scatterplots with surface rendering visualization techniques (such as enclosing surfaces and convex hulls), regarding viewers' performance for several tasks, such as cluster or outlier counting. They found that point-

based scatterplots give a decent overall performance, however nonconvex hulls were preferred most by the participants.

Based on this investigation, another study was conducted to compare viewers' performance in stereoscopic immersive environments and on a non-stereoscopic 2D screen [EML13]. A six-sided immersive virtual reality (VR) system that supports user interactions like zooming, translation and rotation was used. As with Poco's study, participants were presented various cluster visualization techniques (including point-based 3D scatterplots) and had to accomplish several tasks. As a result, surface-based techniques clearly outperformed point-based scatterplots in stereoscopic VR environments, indicating that point-based techniques are inefficient in terms of stereoscopic rendering. Even worse, the hypothesis that 3D projections in a stereoscopic VR environment improve performance on global analysis tasks (such as pattern identification) compared to a 2D screen was rejected; supposedly, this was due to the fact that many viewers had trouble maintaining the global picture when immersed in a VR environment. However, this paper further investigates this topic with special focus on point-based scatterplots.

Accordingly, Ware and Franck [WF96] explored the benefits of presenting abstract data in 3D. For this purpose, a user study was conducted to compare viewers' understanding of graphs when rendered in 2D, in 3D as a static perspective, in 3D stereo, in 3D stereo with motion cues and in 3D stereo with head coupled perspective. As in this work, active shutter glasses were used. The result indicate that stereo viewing increase the size of an abstract graph that can be understood by a factor of 1.6 and even more when head coupled 3D is used. On the other hand, motion cues are considered more significant than stereo cues. Additionally, Gleicher et al. [G*13] showed that the human visual system is able to compare average values in multiclass scatterplots efficiently and accurately, no matter how many points per class or additional distracting classes are added. For his work, he performed a large-scale perceptual study using Amazon's Mechanical Turk, a crowdsourcing Internet marketplace. Participants were not time constrained. On the other hand, Baldassi



Figure 3: A common 2D scatterplot showing multivariate data, whereby red and blue depict clusters of different attributes. Similar scatterplots have been evaluated in the presented study.

et al. [BMB06] showed that perceptual decisions in dense environments can become error-prone when distracting classes are added. Participants not only submitted erroneous answers, they were also highly confident of their decisions. Scatterplots are often dense, which affects the quality of perceptual decisions. Mayorga and Gleicher [MG13] also proposed a way to overcome overdraw in scatterplots. They noted that traditional scatterplots often suffer from heavy overdraw, making it difficult for the viewer to discern data distributions and relationships among clusters. Splatterplots offer a solution by showing dense regions of points as contour-bounded filled areas and subsampling the number of points outside these areas. In addition, GPU-accelerated algorithms were implemented that make it possible to view detailed information in splatterplots by interactive navigation. Unfortunately, because of the fact that splatterplots are based on information abstraction, a loss of information is often inevitable.

Furthermore, Sips et al. [SNLH09] noted that it is often necessary to map high-dimensional data to lower-dimensional views. This is especially true for multi-dimensional scatterplots. Since mapping 3D data to 2D scatterplots can lead to a loss of information, or, even worse, to a misleading representation, it is important to represent the 3D data in such a way that a maximum of features can be perceived by the viewer. Stereoscopic rendering of 3D scatterplots as it is described in this paper is an approach to this topic. At last, Ruvalcaba [Ruv10] performed a small-scale study to determine which representation technique is most efficient to perceive clusters and outliers in high-dimensional data. They compared a single 2D scatterplot, a scatterplot matrix which shows all pairwise scatterplots of the variables in a single view, and a single 3D scatterplot. In detail, participants explored 38 different cluster and outlier tasks in random order. With each task, the participant had to identify clusters and outliers, while the computer program measured time and accuracy. As a result, the performance depended rather on the used dataset than on the representation technique. However, the study indicates that 3D scatterplots can be efficient when clusters overlap whereas 2D scatterplots benefit from well separated clusters.

Stereoscopic Rendering. Like scatterplots, stereoscopic rendering is the subject of many studies. Lo et al. [LC03] conducted user experiments to find out if stereoscopic rendering affects humans' judgement of the realism of computer-generated images. Most of the computer-generated images focus on monocular visual cues such as shading, texture gradients, relative size, occlusion and linear perspective to provide depth information. A group of participants were asked to rate the realism of rendered images, once without stereo, once with stereo vision conditions. The experiment shows that it takes considerably more time to assess the realism of a scene when it is viewed in stereo. Finally, depth cues in computer graphics have been investigated extensively. Many authors deal with the role of color as a monocular depth cue $[B^{*}07, T^{*}91]$. It has been shown that in special situations, color can play a significant role in depth perception (see Section 3.2 for more details). Accordingly, Cleveland and McGill [CM83] have run simple experiments to prove that color can cause optical illusions on statistical graphs. Various test persons were shown a red and green colored map where both regions were the same size. On average, 49% of the test persons were positive that the red region was bigger, 22%

rated the green area to be bigger, and 31% were correct, that is, both regions were the same size. As a conclusion, the use of color on statistical graphs should be treated with great prudence.

Recent studies investigated the impact of stereoscopy on gaming experience. By using a Nvidia 3D Vision system equal to the one utilized in this work, Schild et al. [SLM12] performed user studies to evaluate player experience in computer games. Results showed a less thoughtful and more direct interaction with stereoscopic games.

3. Study Data Generation

The following section explains the implementation details of the software that was used to render 3D scatterplots and perform the study experiment. Subsequently, Section 4 will explain the setup of our user study, before discussing the results.

3.1. Cluster Generation

L

Clusters were generated by drawing a specific number of samples from a multivariate normal distribution. The density function of the latter is given by

$$f(x;\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} \exp(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu))$$

with mean vector μ and covariance matrix Σ given by

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{pmatrix}, \boldsymbol{\Sigma} = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{pmatrix}$$

and a *n*-dimensional random variable **X**, with $X \sim \mathcal{N}_n(\mu, \Sigma)$. The 3×3 covariance matrix Σ must be positive definite [Ton90].

To draw a sample from the multivariate normal distribution, the following approach was used:

- 1. define μ and Σ as zero vector resp. identity matrix (or any other valid vector resp. matrix as described above)
- 2. by using the Cholesky decomposition, find any real matrix *L*, such that $\Sigma = LL^T$
- 3. generate a vector $v = (v_1 \quad v_2 \quad v_3)^T$, with components being independent standard normal distributed values
- 4. the sample $x = \begin{pmatrix} x_1 & x_2 & x_3 \end{pmatrix}^T$ is now given by $x = \mu + Lv$

Steps 2-4 are repeated for every sample in every cluster.

By altering the values of the mean vector μ , it is possible to move the cluster in 3D space. Changing the values of the covariance matrix Σ leads to transformations regarding size, dilation, and rotation of a cluster. Figure 4 shows an example of such a normal distribution visualized as a 3D scatterplot.

^{© 2017} The Author(s) Eurographics Proceedings © 2017 The Eurographics Association.

v. Onzenoodt et al. / Evaluating the Influence of Stereoscopy on Cluster Perception in Scatterplots



Figure 4: A plot of 100 samples drawn from a multivariate normal distribution

3.2. Cluster Rendering

A study by Bailey et al. $[B^*07]$ has shown that color can have a significant effect on the perception of depth. In particular, warmcolored objects (such as red, orange or violet) appear to be closer in depth than cool-colored objects. This effect even strengthens when the background color is dark. Therefore, the background color was chosen to be white. The marked cluser was displayed in orange, whereas the color of the distracting clusters and outliers was cyan. One advantage of these colors is, that they can be perceived by people suffering from red-green color blindness.

While color was not an overriding depth cue, relative size and occlusion was. Unfortunately, in OpenGL, points are not scaled based on their distance to the camera. In fact, the point size remains constant no matter how close or how far away points are rendered. Thus, for the 3D stimuli, a simple scaling function was implemented by using shaders, limiting the point size to 1 pixel minimum for distant points and up to 12 pixels maximum for closer points. Occlusion is simply enabled by calling *glEnable(GL_DEPTH_TEST)* and using the *GL_LESS* depth function. The camera position in 3D was set to -17.5 units into the screen; the camera rotation was set to 18° on the x-axis and -30° on the y-axis. A cartesian coordinate system was rendered with appropriate axis labels (see Figures 1 and 2). Hereby, it was ensured that all participants had the same understanding of the orientation of the coordinate system.

3.3. Active Stereoscopic Rendering

The main goal was to implement active stereoscopic rendering using shutter glasses. However, due to its simplicity and relative cheap realization, anaglyph stereo was implemented for testing purposes at first. In OpenGL, the following approach was taken:

- set up a projection matrix for the left eye
- render scene in red color by using *glColorMask(true, false, false, false, false)*
- set up a projection matrix for the right eye

• render scene in cyan color by using glColorMask(false, true, true, false)

The projection matrices for the left and right eye can be calculated by shifting the camera position and using a function like *gluLookAt* to orientate both cameras to the same focus point. This method for creating stereo pairs is known as **toe-in**. Unfortunately, it results in non-parallel projection planes for each camera.

Thus, it can be uncomfortable for the viewer to look at stereo images created with the toe-in method, especially when objects are not in the center of the screen. The solution to this problem is to use an off-axis projection, where both eyes maintain parallel view directions. Unlike the toe-in method, it requires both frustums to be non-symmetric. Objects in front of the projection plane will appear in front of the screen, whereas objects behind the projection plane will appear behind the screen.

In general, two factors influence the quality of the stereo effect: the distance between the left and the right camera referred to as *eye-separation*, and the distance between camera and projection plane (*focal length*). A common rule of thumb states that the eye-separation should be $\frac{1}{30}$ of the focal length [ZS12, p. 28].

Although the anaglyph 3D method resulted in a clearly visible stereo effect, the quality was not satisfying in terms of color and depth perception. Thus, this method was discarded in favor of active stereoscopic rendering using shutter glasses.

In order to use active stereoscopic rendering, a 3D computer monitor supporting a refresh rate of 120Hz and shutter glasses must be present. In this study, an Asus VG248QE 24 inch monitor and a Nvidia 3D Vision 2 Kit were used, the latter including wireless shutter glasses, an infrared emitter that handles the synchronization between glasses and monitor, and a software driver.

When using DirectX, Nvidia 3D Vision takes the data sent by the application to the stereo driver and renders each scene twice. The stereoscopic 3D monitor then displays the left eye view for even frames and the right eye view for odd frames [Gat09].

Unfortunately, the support of Nvidia 3D Vision for OpenGL is limited. Although the OpenGL standard does support stereo rendering by means of quad buffering, it is not implemented in the Nvidia GeForce Driver, only in the professional Nvidia Quadro Driver. Therefore, an Nvidia Quadro FX 3700 video card had to be obtained. In addition, a 3-pin mini-DIN stereo connector linked the GPU and the infrared emitter, ensuring synchronization regardless of current CPU load, which lead to a flicker-free 3D experience.

With the Nvidia Quadro graphics card, quad buffer stereo could be implemented. The projection matrices are set up following the same procedure as previously described. Again, the scene is rendered twice, once into the back left color buffer and once into the back right color buffer.

Since the application had to run at 120 frames per second minimum, special attention has been paid to optimization. However, this was not a problem since the rendered geometry was small and inexpensive.

4. User Study Setup

The following section outlines the design of the user study conducted to investigate the perceptual implications of different scatterplots. In addition, hypotheses about users' performance are formulated.

4.1. Experimental Design

The goal of the experiment was to find out to what extend stereoscopic rendering helps the viewer to perceive clusters in pointbased scatterplots. Therefore, a computer program was implemented, to generate and display a set of scatterplots in randomized order. Each set consised of three types of scatterplots: 2D scatterplots, 3D scatterplots (see Figures 1 and 2), and 3D stereoscopic scatterplots, with the latter being the same as their 3D counterparts but rendered in stereo. 2D scatterplots were generated by orthographic projection from their 3D counterparts, removing the depth component. Regarding the 3D stereoscopic scatterplots, position, rotation, and size of the plots remained constant within each set, but varied between sets.

In order to validate the experimental setup, evaluate the overall 3D experience, and collect initial feedback on the 3D environment, a pilot study was carried out with three participants. The participants were undergraduate and graduate computer science students, all of whom were familiar with scatterplots. The timeout for answers was adjusted as well as the complexity of the scatterplots which were perceived (stimuli).

Following the pilot study, a second study with a larger number of participants was carried out. At the beginning of each experiment, participants were asked to fill out a questionnaire. Questions about their experience with stereoscopy and scatterplots were asked, and information about the presence of visual impairments, such as myopia, hyperopia, or red-green color blindness were gathered.

Participants were then handed the active shutter glasses and instructed to wear them permanently during the experiment. Additionally, they were asked to move their head as little as possible, to ensure the least possible difference in the viewing conditions for each stimulus. This is important since the computer display appears darker when wearing shutter glasses and the viewed image can be distorted when viewed at an angle. Furthermore, participants were instructed to work as fast as possible, as a timer was running down.

Each participant was shown a set of simple test stimuli, giving them the possibility to get accustomed to the environment. Participants who misjudged more than 50% of the 2D positions of the marked clusters, were barred from the rest of the study. However, this was never the case. In total, 30 stimuli were presented in randomized order to each participant (10 sets, each consisting of a 2D scatterplot, the same scatterplot in 3D with depth, and in 3D with depth and stereoscopy). Each of these sets consised of four sparse and six dense scatterplots. The computer program randomly chose one of the axis (X, Y or Z in 3D; X or Y in 2D) and generated three answer options for the position, one of which was correct (see Figure 5). The participant was then asked to assess the center of a cluster marked by color. By using this method, the problem of giving the exact position of a cluster was simplified by choosing between three possible answers, demanding as little effort as possible.

Participants had to choose an answer within 25 seconds, giving them enough time to examine the scatterplot. The time limit was set, since studies have shown that participants are able to rapidly judge multi-class scatterplots, but tend to take more time for it [LMvW10b,LMvW10a,EML13]. Furthermore, the study was designed to examine the correctness of participants' choices rather than the time they needed to decide. However, when they did not respond within the 25 seconds limit, the stimulus was skipped.

At the end of each experiment, participants were asked whether they think their depth perception improved when scatterplots were rendered in stereo.

4.2. Hypotheses

It is hypothesized that the overall performance is best with 2D scatterplots and worst with 3D scatterplots; we hypothesize further that stereoscopic rendering should be located inbetween these two.

The hypothesis is supported by various studies [PEP*11, EML13]. An investigation showed that stereoscopic 3D displays can improve spatial perception for tasks that are multi-dimensional in nature, tasks that are difficult and unfamiliar, and tasks that lack other spatial visual cues [ML14]. However, as mentioned before, point-based scatterplots did not perform as well as other visualization techniques when rendered in stereoscopic 3D. Additionally, binocular disparity may not necessarily be the decisive depth cue. Hence, the following hypotheses are formulated:

- **H1** Performance will be lower with 3D scatterplots when compared to 2D scatterplots.
- **H2** Performance will be lower with 3D non-stereoscopic rendering when compared to 3D stereoscopic rendering.
- **H3** Overall performance will be higher when perceiving sparse scatterplots when compared to dense scatterplots.

While hypotheses H1 and H2 are concerned with performance in regards to stereoscopic rendering, hypothesis H3 addresses the issue of the perception of dense and sparse scatterplots. In this case, dense scatterplots are diagrams which show more than one class and at least 50 distracting points, while sparse ones show either one or two clusters without any noise or a single cluster with a maximum of 100 distracting points.

🔻 An	swers	:	
Time	e lefi	t: 14 sec.	
Cent	er of:	f orange cluster is at	
y	=	0.0	
y	=	-1.0	
y	=	-2.0	

Figure 5: GUI showing an example for the answer options presented to the study subjects.

© 2017 The Author(s) Eurographics Proceedings © 2017 The Eurographics Association. v. Onzenoodt et al. / Evaluating the Influence of Stereoscopy on Cluster Perception in Scatterplots



Figure 6: Performance of participants grouped by stimuli (2D, 3D and stereoscopic 3D). Times were measured per stimulus.

5. Experiment & Discussion

In this section, the user experiments are outlined and the results are discussed on the basis of statistical values.

5.1. Experiment

The main study took place during one week in a science laboratory at our university. A total of ten students participated, all with a background in computer science. Seven participants were male, three female, the age ranged from 19 to 25. Four out of ten participants were familiar with scatterplots and had used them in a scientific context before; only four participants had experience with stereoscopy. All participants had normal or corrected-to-normal vision, none of the participants were color-blind or had achromatopsia.

Participants practiced with several sample stimuli until they felt familiar with the task. Afterwards, as described in Section 4.1, they had to assess the center of 30 clusters within a 25 seconds time limit for each stimulus. None of the participants exceeded the time limit. Each session lasted approximately ten minutes. After each session, the participants were asked whether they felt that stereoscopic vision affected their perception of depth in a positive manner.

5.2. Results

The diagrams shown in Figure 6 and 7 show the mean value of the success rate for each type of scatterplot in percent, as well as boxplots of our time measurements.

The study showed that viewers performed best with 2D scatterplots; 93% of all cluster positions were assessed correctly. Performance decreased with 3D stereo scatterplots, where 79% of the answers were correct. However, performance was worst with 3D non-stereo scatterplots; only 74% of all cluster centers were perceived correctly by participants. These results are summarized in Figure 6.

Furthermore, the results indicate that viewers tend to take more time to judge the center of a cluster in a 3D scatterplot, which was already shown by Etemadpour et al. [EML13]. However, only a minor difference could be observed between stereoscopic and nonstereoscopic 3D scatterplots. These results are also visualized in Figure 6.



Figure 7: Mean correctness of dense and sparse scatterplots.

Additionally, participants' performance when exposed to sparse scatterplots in comparison to dense scatterplots (plots with a higher number of clusters, and distracting classes) should be further investigated. As shown in Figure 7, stereoscopic rendering increases performance in sparse scatterplots.

Six of the participants felt that stereoscopic rendering slightly improved their ability to perceive depth, while the remaining four did not notice any effect at all.

5.3. Discussion

In this section, the results of the user study are discussed with reference to the hypotheses formulated in Section 4.2.

Even though the results are not significant, they show a tendency towards the expected differences. A study with a greater number of participants might lead to significant differences, further strengthening the support for our hypothesis.

As predicted, the perception of clusters was best in 2D scatterplots, supporting hypothesis **H1**. Most participants had no problem assessing the center of a cluster in 2D. This might be due to the fact that 2D scatterplots are easy to perceive by nature. However, the level of difficulty increased, when marked clusters had low density and a high spread. This indicates that clusters which are dense and coherent yield a better recognition rate in 2D. In addition, the number of correct answers was almost equal in dense and sparse scatterplots.

The results also support hypothesis **H2**. The user study showed that there was a subtle improvement when using stereoscopic rendering. However, the effect was less prominent, than initially suspected. This indicates that stereoscopic rendering does have an influence on the perception of clusters in scatterplots, when point-based rendering techniques are used. However, as seen in a study by Etemadpour et al. [EML13], performance should considerably increase when using surface-based techniques.

The comparison of sparse and dense scatterplots supports hypothesis **H3**. The results show better performance for sparse scatterplots, when compared to dense ones. Stereoscopic rendering seems to be especially helpful for sparse scatterplots.

It can be concluded that point-based scatterplots benefit from stereoscopic rendering, even though the effect is not as strong as initially suspected. This may be attributed to the fact that clusters rendered with point primitives are incoherent geometric objects and therefore difficult to perceive; using surface-based rendering techniques could considerably increase performance, as shown in other studies [PEP*11, EML13].

6. Conclusion

A user study was conducted to investigate the influence of stereoscopic rendering on the perception of clusters in point-based scatterplots. A typical cluster analysis task was performed by a set of participants. They were presented with 2D scatterplots, 3D scatterplots and stereoscopic 3D scatterplots. Participants had to choose the center of a cluster marked in one color from a set of three possible answers. The results indicate that stereo rendering improves performance, especially with sparse scatterplots.

While our study shows a tendency towards the expected results, our number of participants was small. Future studies with a larger number of participants could be conducted to further strengthen our hypotheses. Future work could also include a study in which additional depth cues, such as motion, or user interactions are implemented and investigated. In doing so, the effect stereo rendering has on the perception of clusters in point-based scatterplots could be further investigated. In addition, an increase in performance with stereoscopic 3D scatterplots was observed, when participants repeated the experiment. This indicates that once users are familiar with the environment, they find it less difficult to perceive clusters. Another user study could be conducted to support this hypothesis empirically. Additionally, the influence of colors on cluster perception in scatterplots could be investigated, since this has not been taken into account in our experiment.

Acknowledgment

We would like to thank Anke Huckauf for discussing the setup of the conducted user study.

References

- [Asi83] ASIMOV D.: The grand tour: A tool for viewing multidimensional data. SIAM J. Scientific and Statistical Computing (1983). 1
- [B*07] BAILEY R., ET AL.: The effect of object color on depth ordering. 3, 4
- [BMB06] BALDASSI S., MEGNA N., BURR D. C.: Visual clutter causes high-magnitude errors. *PLoS Biology* 4 (3) (2006). 3
- [CM83] CLEVELAND W. S., MCGILL R.: A color-caused optical illusion on a statistical graph. *The American Statistican* 37 (2) (1983). 3
- [EML13] ETEMADPOUR R., MONSON E., LINSEN L.: The effect of stereoscopic immersive environments on projection-based multidimensional data visualization. *Information Visualisation (IV), 2013* 17th International Conference (2013). 2, 5, 6, 7
- [FD05] FRIENDLY M., DENIS D.: The early origins and development of the scatterplot. *Journal of the History of the Behavioral Sciences* 41 (2) (2005). 1
- [G*13] GLEICHER M., ET AL.: Perception of average value in multiclass scatterplots. *IEEE Trans. on Visualization and Comp. Graphics 19 (12)* (2013). 1, 2
- [Gat09] GATEAU S.: The in and out: Making games play right with stereoscopic 3d technologies. *Game Developers Conference* (2009). 4

© 2017 The Author(s)

- [LC03] LO C., CHALMERS A.: Stereo vision for computer graphics: The effect that stereo vision has on human judgements of visual realism. SCCG '03 Proceedings of the 19th Spring Conference on Computer Graphics (2003). 3
- [LMvW10a] LI J., MARTENS J.-B., VAN WIJK J. J.: A model of symbol lightness discrimination in sparse scatterplots. 2010 IEEE Pacific Visualization Symposium (2010). 1, 5
- [LMvW10b] LI J., MARTENS J.-B., VAN WIJK J. J.: A model of symbol size discrimination in scatterplots. *Proceedings of the 2010 ACM annual conference on Human Factors in Computing Systems (CHI 2010)* (2010). 1, 5
- [Meh13] MEHRABI M. O.: Making 3d work: A classification of visual depth cues, 3d display technologies and their application. *Research and Practice in Information Technology 139* (2013). 1
- [MG13] MAYORGA A., GLEICHER M.: Splatterplots: Overcoming overdraw in scatter plots. *IEEE Trans. on Visualization and Comp. Graphics* 19 (9) (2013). 3
- [ML14] MCINTIRE J. P., LIGGETT K. K.: The (possible) utility of stereoscopic 3d displays for information visualization: The good, the bad, and the ugly. *IEEE VIS International Workshop* (2014). 5
- [PEP*11] POCO J., ETEMADPOUR R., PAULOVICH F., LONG T., ROSENTHAL P., OLIVEIRA M., LINSEN L., MINGHIM R.: A framework for exploring multidimensional data with 3d projections. *Euro-Vis'11 Proceedings of the 13th Eurographics* (2011). 2, 5, 7
- [Ruv10] RUVALCABA J. D.: Comparing a single 2d scatter plot, a single 3d scatter plot with motion, and a 2d splom for reduced multidimensional data. 3
- [SLM12] SCHILD J., LAVIOLA J., MASUCH M.: Understanding user experience in stereoscopic 3d games. CHI '12 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (2012). 3
- [SNLH09] SIPS M., NEUBERT B., LEWIS J. P., HANRAHAN P.: Selecting good views of high-dimensional data using class consistency. *Euro*graphics / IEEE-VGTC Symposium on Visualization 28 (3) (2009). 3
- [SW09] SANFTMANN H., WEISKOPF D.: Illuminated 3d scatterplots. Eurographics/IEEE-VGTC Symposium on Visualization 28 (3) (2009). 2
- [T*91] TRSOCIANKO T., ET AL.: The role of colour as a monocular depth cue. Vision Research 31 (11) (1991). 3
- [Ton90] TONG Y.: The Multivariate Normal Distribution. Springer Publishing, 1990. 3
- [WF96] WARE C., FRANCK G.: Evaluating stereo and motion cues for visualizing information nets in three dimensions. ACM Transactions on Graphics 15 (2) (1996). 2
- [ZS12] ZHAO M., SHA J.: Communications and Information Processing. Springer Publishing, 2012. 4

Eurographics Proceedings © 2017 The Eurographics Association.

Concepts of Hybrid Data Rendering

T. Gustafsson, W. Engelke, R. Englund and I. Hotz

Linköping University, Department of Science and Technology, Media and Information Technology.

Abstract

We present a concept for interactive rendering of multiple data sets of varying type, including geometry and volumetric data, in one scene with correct transparency. Typical visualization applications involve multiple data fields from various sources. A thorough understanding of such data often requires combined rendering of theses fields. The choice of the visualization concepts, and thus the rendering techniques, depends on the context and type of the individual fields. Efficiently combining different techniques in one scene, however, is not always a straightforward task. We tackle this problem by using an A-buffer based approach to gather color and transparency information from different sources, combine them and generate the final output image. Thereby we put special emphasis on efficiency and low memory consumption to allow a smooth exploration of the data. Therefore, we compare different A-buffer implementations with respect to memory consumption and memory access pattern. Additionally we introduce an early-fragment-discarding heuristic using inter-frame information to speed up the rendering.

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

1. Introduction

In this paper we discuss methods to combine different rendering concepts for volumes, meshes and texture based techniques for the purpose of interactive hybrid visualizations.

Visualization plays an important role for the analysis of data in many scientific applications. Thereby, typical real world scenarios involve multiple data sets often of different type. This requires efficient solutions for combined rendering of hybrid data in one image. Since different data types often require fundamentally different rendering techniques this can be a challenging task; e.g. the joint visualization of geometry represented as multiple transparent surfaces and volumes using direct volume rendering.

When combining multiple transparent surfaces it is important to blend them in the right order to be perceived correctly. This means the fragments need to be composited in the correct order, either front-to-back or back-to-front. In direct volume rendering, when using volume ray-casting, volume samples are composited in a similar way, in order from the entry point to the exit point. Since volume ray-casting is working on a range of samples at once it is impossible to use the default compositing methods that exists on the GPU to combine volume ray-casting with transparent surfaces. *Order Independent Transparency* refers to a collection of techniques to render multiple, possibly complex, transparent surfaces independent of the order of draw calls. Most of the Order Independent Transparency techniques focus on surface representations rather than combining surfaces and volumetric data.

In this paper we will discuss concepts extending existing Order Independent Transparency techniques for interactive hybrid data rendering. In detail we consider

- · Mesh rendering for geometry visualization
- · Volume Ray-casting for direct volume visualization
- 3D line integral convolution (3D-LIC) for directional data.

Our approach is based on A-buffer techniques and we demonstrate the effectiveness of this on datasets from different fields including a heart dataset containing both anatomical context in the form of surface boundaries and the blood flow, as well as a protein dataset containing various surface representations and the electron charge density volume. With this we present the following contributions:

• A concept for efficient hybrid data rendering based on A-Buffer techniques. • An improvement over existing techniques by using interframe information for a fragment discard heuristic.

The remainder of this paper is organized as follows. After related work in Section 2 we introduce some important background knowledge in Section 3. We present the methods of our approach in Section 4. In Section 5 we analyze timing and memory consumption and present our results. Finally, we conclude our work with a discussion in Section 6 and name possible future developments.

2. Related Work

In the following, we shortly review the most relevant literature for our work. This is at first order independent transparency, basic volume rendering and three dimensional texture rendering.

ORDER INDEPENDENT TRANSPARENCY. The correct rendering of (semi-) transparent scenes is an intensively studied field in computer graphics. Early techniques were based on sorting the individual draw calls. Opaque objects are drawn first, followed by (semi) transparent objects. With increasing geometric complexity and an increasing amount of objects in typical computer graphics scenes, this technique leads to high computational costs for the CPUbased sorting of scene elements in a back to front manner. Nvidia [Eve01] presented an approach, which uses consecutive peeling of the depth buffer to resolve correct transparency, independent from the order of draw calls. The individual slices of the depth buffer reveal hidden geometry in a consecutive manner. All slices are blended back-to-front in an additional render pass. This method was later improved by Bavoil and Myers [BM08] where a front-to-back and back-to-front peeling is performed at the same time.

Another approach to render correct transparency, independent of the order of draw calls, is sorting on fragment level. For this, all objects of a scene are rendered and their color and depth values are stored in a buffer. This A-Buffer approach can be implemented with layered 2D textures or per-fragment linked lists [BK11]. Maule et al. and Zhang [MCTB12, Zha14] have addressed different performance and memory related aspects of this approach. One challenge with the A-buffer based methods is the optimization of the local memory for each pixel's fragment list. For example, GPU shaders do not allow the usage of arrays with dynamic size. An early work by Sintorn et al. targeted this issue by a rapid pre-computation of the required array size [SEA08]. Later, Lindholm et al. [LFS*14] introduced a different approach. Their method is called *per-pixel array* optimization and is based on using multiple shaders which use different array sizes, and sending pixels of certain depth complexities to a shader using a similar array size. Lindholm also proposes another improvement called per-pixel depthpeeling, which removes the problem of having to allocate too large array sizes completely. Schollmeyer et al. presented the integration of an A-Buffer approach with a deferred rendering pipeline [SBF15].

Another method, more similar to the traditional depthbuffer, was presented by Bavoil et al. [BPL*07]. Their *kbuffer* is also capable of handling order independent transparency. It stores the front-most fragment for each pixel, but it can only store up to *k* fragments, sorted and blended in a single pass. One advantage of this, compared to the A-buffer, is that it does not have to define a maximum scene depth.

DIRECT VOLUME RENDERING. Volume raycasting is the most intuitive and popular method for direct rendering of volumetric data with scalar quantities. Early work in this area was undertaken by Cabral et al. [CCF94] where they used 3D texture slicing. More recent approaches render proxy geometry to generate entry and exit point textures on the GPU and perform ray casting afterwards. These approaches have been improved by Krüger and Westerman [KW03] by integrating early ray termination and empty space skipping. Röttger et al. [RGW*03] targeted preintegration technique, volume clipping and advanced lighting. A flexible framework for standard and non-standard techniques was presented by Stegmaier et al. [SSKE05]. Their work was easy to extend and able to reproduce translucency, transparent isosurfaces, refraction and reflection. Volume rendering for general polyhedral cells was presented by Muigg et al. [MHDG11] The Visualization Handbook [KM05] as well as Real-Time Volume Graphics [EHK*06] provide an overview of direct volume rendering techniques.

LINE INTEGRAL CONVOLUTION. Dense texture based techniques are one major group in flow visualization. One representative of this category is line integral convolution which was first presented by Cabral et al. [CL93]. Stalling and Hege [SH95] presented a fast and resolution independent approach. Later, the idea was applied to 3D vector fields by Interrante [Int97] as well as Rezk-Salama et al. [RSHTE99] and surfaces by van Wijk et al. [vW03]. A comprehensive overview can be found in the work of Laramee et al. [LHD*04]. Falk and Weiskopf [FW08] targeted 3D line integral convolution in conjunction with direct volume rendering. Additionally, they incorporated adaptive noise generation. Therefore, the method is independent of the input data size.

3. Background

In this section we will briefly describe the visualization methods we focused on for the combination with our hybrid data rendering approach.

MESH RENDERING. A typical approach for mesh rendering is to transform the meshes' vertices with the current modelview matrix, perform primitive assembly, clipping, perspective division, depth testing and finally put each fragment to the output buffer. OpenGL performs these stages



Figure 1: Our hybrid rendering approach combines rendering of mesh and volume data into one single visualization. The shaded fragments of the mesh rendering (a) are stored with their depth information in an A-Buffer. During ray traversal the volume renderer (b) stores its results in the same A-Buffer. The result (c) is a combined visualization of both source with correct transparency.

in its configurable and customizable pipeline approach. In our current implementation we rely on triangulated meshes. We suggest [SSKLK13] for further reference.

VOLUME RAYCASTING. For direct volume rendering, a common method is volume ray-casting. Here, a three-pass approach can be used. In the first pass a proxy geometry, namely the bounding box of the volume, is rendered. During this pass, back-face culling is enabled and a color buffer is bound as render target. Its content will later serve as entry-point texture. For the second pass, front-face culling is enabled and therefore the exit-point texture is generated. Both passes will interpolate the given texture coordinates at each vertex of the bounding box in an efficient way. Afterwards, the third and last pass uses the entry and exit-point textures to define the rays used for sampling the volume. We suggest *The Visualization Handbook* [KM05] and *Real-Time Volume Graphics* [EHK*06] for further reference.

3D LINE INTEGRAL CONVOLUTION. Line Integral Convolution (LIC) is a well known texture based method for visualization of steady vector fields. The basic idea is to perform a convolution of a white noise input texture. At each location in the field a one-dimensional kernel is defined by integrating a flow tangential line a fixed distance in both, forward and backward direction. The noise texture is sampled along this line and the corresponding values are summed and normalized. Afterwards, the resulting value is placed at the actual location in the output image. Furthermore, the local one-dimensional nature of this algorithm makes it ideal for a parallel and efficient implementation. Today, LIC calculations are commonly performed on graphics hardware and can easily achieve interactive frame rates even for larger input data.

For 3D vector fields the same approach can be used. In-

stead of a two-dimensional noise texture, a noise volume is utilized to perform the convolution at each voxel. Afterwards, the result is visualized by direct volume rendering. Unlike 2D LIC, 3D LIC is more sensitive to the type of noise used for the convolution. A sparsely populated and blurred 3D texture as input has proven to result in less cluttered results. Figure 2 shows an analytically defined vector field visualized by streamlines as well as 3D LIC. We suggest [CL93] and [FW08] for further reference.



Figure 2: Visualizations of an analytically defined vector field using (a) streamlines and (b) 3D LIC.

NUMERICAL INTEGRATION METHODS. For the calculation of integral lines typical methods are Euler and Runge-Kutta integration. Both methods iteratively compute the integral lines from a given start location. Thereby, Euler integration samples the field only at the beginning of the interval (e.g., current location) to compute the next point and thus suffers from low accuracy. In contrast, the fourth-order Runge-Kutta scheme (RK4) samples the field at four different locations within the current interval and uses a linear combination of these samples to compute the next point. The

T. Gustafsson & W. Engelke & R. Englund & I. Hotz / Concepts of Hybrid Data Rendering



Figure 3: A screen texture is used to store pointers to the last stored fragment, which in turn points to its previous fragment. By using these pointers, we are able to access the RGBA+depth values of each fragment.

RK4 integration is a good trade-off between computational cost and accuracy.

4. Hybrid Data Rendering

In this paper we propose a concept to combine different rendering techniques, this is visualizing volumes using *direct volume rendering* together with (semi-) transparent geometry. For a combination of mesh rendering and volume raycasting it is important to align the individual coordinate systems. Doing this ensures that the resulting depth values correspond to each other.

In this section we describe the proposed hybrid data rendering method. This includes the data types that can be visualized and the visualization options.

A-BUFFER RENDERING METHODS. Order independent rendering of (semi-) transparent scenes is commonly done in two rendering passes. The first pass renders the scene and populates the A-Buffer with fragments, the second pass sorts these fragments and performs back-to-front compositing to produce the final pixel colors. In order for our A-Buffer to support varying color and transparency of fragments and be able to correctly compose these, we need to be able to store the color, its transparency, and the depth value. This results in five different values in total which have to be stored per fragment (r, g, b, a, depth). Therefore, we are using two buffers, one for (r, g, b, a) and one for depth values.

An A-Buffer can be implemented with different underlying technologies. The most intuitive approach is storing each rendered fragment in a layered 2D-texture. All layers, for one output fragment, are later blended in the order of their depth value. A major drawback of this approach is that it suffers from a lot of unused memory if the depth complexity varies over the scene.

Another approach uses a *per-pixel linked lists* to store the incoming fragments. This approach requires a single buffer to store all fragments, with an accompanying buffer that stores pointers between these elements. Thereby, the first fragment of each coordinate points to the zero index, all other fragments point back to its previous fragment. Thus,



Figure 4: *Here the screen texture points to a page of fragments. In this illustration the page size is three fragments, but it can essentially be any number. By storing the fragments in pages, we get better performance on the GPU.*

a lot of memory can be saved compared to using the texturebased method since the buffer can be defined to be just as large as necessary. This is advantageous especially for scenes with high variation of depth complexity. Figure 3 shows the concept of using linked lists of fragments with an A-Buffer. In general, this method is slower than a texturebased method, due to bad cache coherency between the fragments, since each fragment is possibly stored in different memory pages.

To overcome this disadvantage we propose to use *fragment pages*, instead of individual unrelated fragments, in the linked list. A pointer in the list will then point to a page of fragments instead of a single fragment. This method minimizes the problem of bad cache coherency when using a linked list of fragments. Figure 4 shows how the pages are stored in the linked list.

All described methods require memory to be pre-allocated on the GPU. For the texture-based method this size is determined by a user-configurable parameter adjusting the size of the texture stack, i.e. number of depth layers per fragment. For the link-list methods the amount of memory to pre-allocate is defined globally and not per pixel, the size of this buffer is automatically adjusted during run-time. Upon loading a scene a relatively small buffer is allocated. If this buffer is fully filled up while rendering we abort the current frame, enlarge the buffer, and restart the rendering.

PERFORMANCE ANALYSIS. Out of the described methods we expect the texture method to be faster than the linkedlist method due to less cache misses. A problem with the texture-based method is that it must predefine a maximum depth complexity that it will be able to handle. The linked lists method can theoretically handle any depth-complexity until the pre-allocated memory is exhausted. The advantage with using this method though is that we will have the option to allocate less memory and still get correct results, compared to a texture-based approach, where the size of the stack must be at least as big as the highest depth complexity. According to Crassin [Cra10], the advantage in memory utilization of the method using linked-lists can be huge when the rendered scene has high variation in depth complexity. In typical scenes, exactly this is the case and is reflected by the values measured by [Cra10]. He found that a linked-list based implementation consumes, depending on the screen resolution, only 6 - 10% of the memory a texture-based approach does.

DIRECT VOLUME RENDERING WITH A-BUFFERS. Volume ray-casting is the most common approach for direct volume rendering, and by recognizing the similarity of the compositing part of our A-Buffer rendering and volume raycasting we realize that to combine these two methods is similar to the merging steps of merge sort and we only need to modify our compositing step. After we have retrieved and sorted the fragments stored in the A-Buffer for current pixel, we query the entry and exit point and perform the raycasting. For each step in the ray-casting loop we compare the depth of current volume sample with the foremost fragment in the sorted A-Buffer list and pick whichever is nearest. This is done until either both the ray-casting and A-Buffer fragment list is empty or we reach full opacity.

EARLY FRAGMENT DISCARDING. In many scenes, there are a lot of fragments that are stored unnecessarily. When the alpha values of the closest fragments are high enough to completely occlude the fragments behind them we would optimally want to have a way of knowing beforehand which fragments those would be, since that would allow us to skip the occluded fragments completely in the first pass.

To address this issue, we present a method we call *early fragment discarding*. This technique relies on an additional buffer and a general assumption of frame coherency. The additional buffer is used to store the depth values of previous frame's fragments where full opacity was reached. With this we do a depth-test for each fragment and discard that fragment if its depth is higher than the depth value of the previous frame. This happens during the first rendering pass, while populating the A-Buffer.

By storing fewer fragments we see performance increases for multiple reasons. The first reason is that discarded fragments does not need to be written to the buffer, which is one of the bottle necks of the first pass, second reason is, having fewer fragments, sorting them in the second pass will become faster.

This approach can introduce information loss during interaction. For example, when rotating the camera the depth values close to edges might change rapidly resulting in discarding fragments that could have a strong contribution to the final pixel-color. By only discarding fragments while the user is interacting with the scene (rotation, translation etc.) and perform a full rendering once the user stops interacting, a good compromise between performance and correctness can be achieved. A way to decrease information loss can be achieved by introducing an allowed margin of depth values. Since the fragment depth is stored in the range of 0-1, we can add a percentage margin to the threshold, which will allow us to keep some of the fragments that could potentially be incorrectly discarded on interaction of the scene. A good value for this margin was found at 5%, in which most of the correct fragments were kept, with still significantly improved performance, compared to not using the method at all.

3D LINE INTEGRAL CONVOLUTION. A 3D-LIC method was implemented using Euler integration, as well as fourthorder Runge-Kutta integration with a fixed step size. For every voxel in a vector field volume, a generated noise volume is traversed with a certain amount of steps, in forward- and backward directions. For every step traversed the noise texture is sampled. After sampling all points along one integral line, their mean value is used as the final color of the current voxel.

RANDOM NOISE-VOLUME GENERATION. Unlike 2D LIC, with 3D LICs the output quality is highly dependent on the chosen strategy for random noise generation. In this case, using a 3D volume as the noise input, different types of noise were tested. These were *white noise*, where each voxel get a random value between zero and one, and a sparse noise generation based on the *Halton sequence*. As it can be observed in Figure 5, the sparse noise based on the *Halton sequence* is the most promising one when comparing the final output images with respect to visual clutter and structure.



Figure 5: Two LIC visualizations of the same vector field, with different noise inputs. White noise (a) give a dense but more cluttered result. Sparse Noise based on the Halton sequence (b) results in clearer, more distinct structures.

5. Results and Evaluation

The methods described in Section 4 have been implemented in Inviwo, an open-source framework for interactive visualization [SSK*15]. In this section we present the results from applying the described methods on two datasets from different domains.

The first dataset is a flow dataset of a human heart, acquired using 4D Flow MRI, and consists of a time-sequence of volumes given over one cardiac cycle. In addition to the flow field we have a time-sequences of the anatomy as regular MRI and segmented masks for each of the various chambers, major arteries and veins of the heart [BPE*15]. In fig-



Figure 6: Visualization of the protein human carbonic anhydrase II. (a) and (b) show two different mesh representations, in (a) ribbons are used to visualize the backbone or the protein and in (b) a licorice visualization is used to visualize the bounds between atoms. Figure (c) visualizes the electron charge density inside the protein using volume ray-casting. By using our hybrid renderer we can combine these three views into the final image, seen in (d). The depth complexity of the A-Buffer of this frame is visualized in (e).

ure 7 we can see various representations of this dataset, created with our software. In Figure 7(a) we have extracted an iso-surface to represent the boundary of the heart, here rendered with full opacity. To visualize the blood flow we have generated a 3D LIC volume as described in Section 4. The result of this is visible in Figure 7(b). When having only the 3D LIC rendering it can sometimes be tricky to depict the anatomical location of the flow in the heart. In such cases rendering the flow together with the surface can be useful. With our hybrid renderer we can easily combine this two representations, in Figure 7(c) we can see how the 3D LIC is rendered using volume ray-casting together with the surface mesh. Before compositing, the surface mesh is rendered into the A-Buffer with a 40% opacity.

The second dataset is the human carbonic anhydrase II [PDB] protein. It consists of the protein structures as meshes and volumetric data describing the electron charge density at each location. In Figure 6(b) we see a Licorice representation of the bonds between atoms in the protein. This representation is available from the data as a polygon mesh of cylinders. In Figure 6(a) a second mesh representation is available which represents the backbone of the protein using ribbons. Overlapping with the meshes, we have an electron charge density volume describing positive and negative charges. This volume is rendered using regular volume ray-casting, Figure 6(c). Red areas represents positive charges and blue areas represent negative charges. Using our hybrid renderer we can combine these three representations. First, the Licorice mesh is rendered into the A-buffer at full opacity together with the ribbons at 60% opacity. The sorted a-buffer is then combined with the volume ray-casting resulting in a final rendering as can been seen in 6(d). In Figure 6(e) the depth complexity of the scene is displayed. This is done by coloring the pixel based on the number of fragments for that pixel where black means zero fragments and red means max number of fragments.

In scenes with very large depth-complexity, the A-buffer

may sometimes run out of memory. In such situations we discard additional fragments. Since discarding fragments during the population of the A-Buffer leads to insufficient information during the final composite step we show black pixels instead of incorrect information at that location.



Figure 7: Heart dataset: (a) shows the extracted surface as geometry, whereas (b) shows the result of the 3D LIC. In (c) both visualizations are combined. Note that the opacity of the geometry was reduced to reveal the inner LIC visualization.

PERFORMANCE. The A-Buffer methods presented in this paper are based on performing two rendering passes. The first pass writes the fragments to a fragment container (*which may be different depending on the method used*), then it renders them to screen in the second pass. When measuring the performance of the A-Buffer methods, it is therefore relevant to measure the performance of these passes separately.

All performance measurements were done in a setup similar to the heart rendering described above. Instead of using 3D LIC, the flow is visualized using path lines rendered as tubes and the anatomical surface rendered using volume raycasting. All performance metrics presented in the paper have been measured on a computer using a Nvidia GTX 580 GPU, with 4 GB of graphics memory, and an Intel Xeon W3550 CPU, with four cores and clock speed of 3.07 GHz.

In order to get a good time measure of each pass, timestamps were taken before the first pass, after the first pass, and after the second pass. By calculating the differences between these we could measure the time taken for each pass. When running the tests using different options meant to speed up one of the rendering passes, it was discovered that both time differences changed drastically, in contrast to just one, which would have been expected.

In the second pass, insertion sort is used to sort all the fragments for each screen pixel, which has a time complexity of $O(N^2)$. Since most pixels do not have a depth-complexity of higher than 10 fragments for most scenes an $O(N^2)$ sorting algorithm is sufficient. Not much, if any, performance could be gained by changing the sorting method into an $O(N \log N)$ such as quick or merge sort. Disabling depthsorting completely improves overall performance by around 20-30%, but will of course give incorrect results.

In Table 1 we present the rendering times for the two A-Buffer methods with two parameters (e.g., max depth and page size). The main difference between the methods is how the GPU memory is used. The texture-based approach uses layered 2D-textures during the population of the A-Buffer. In contrast the linked list approach uses pointers in GPU memory for each fragment location (see Figure 3, 4). Section 4 describes both methods in detail. The image size for these tests was 800x800 pixels. Note that the early-fragment-discarding method was not used here.

The early-fragment-discarding method was tested with scenes of varying depth complexity, object transparency, and resolutions. The results are shown in Table 2.

Here it is clear that the method is able to improve performance of the A-Buffer significantly under the right circumstances. An unexpected result was obtained for low resolutions, where removing the margin yielded worse performance compared to using a 5% margin.

Table 1: Performance of the two A-Buffer methods. The average depth-complexity was ~ 12 fragments for the scene these tests were made on.

Method	Max Depth	Page Size	Avg. FPS
Texture Based	64	-	~22
Texture Based	128	-	~ 18
Linked Lists	64	8	~18
Linked Lists	128	8	~16
Linked Lists	128	2	~15
Linked Lists	128	1	~14

Table 2: Performance of the Early Fragment Discarding method using different parameters. All tests were performed on the same dataset. The average depth-complexity of the scenes were ~ 10 fragments for the 400x400 px scene and ~ 7 for the 800x800 px scene.

Dimensions	Alpha	EFD	Margin	Avg. FPS
400x400	0.2	no	-	~30
400x400	0.2	yes	5%	~35
400x400	1.0	yes	0%	~44
400x400	1.0	yes	5%	~46
400x400	1.0	yes	10%	~43
800x800	0.2	no	-	~ 24
800x800	0.2	yes	5%	~32
800x800	1.0	yes	0%	~41
800x800	1.0	yes	5%	~ 40
800x800	1.0	yes	10%	~39

Using 10% margin in general removed the visual artifacts. In our examples, a margin of around 5% was a good compromise resulting in few artifacts and generally high image quality. Removing the margin generally increases performance, however it has a large impact on the image quality which is considered as too big. Since the performance gain is small using margins is justifiable.

6. Conclusion and Future Work

CONCLUSION. With this work we presented a hybrid rendering technique which is capable of combining different visualization methods into a single output image in an efficient way. To achieve a high-quality and interactive rendering we extended and combined state of the art methods like the A-Buffer and GPU based volume ray-casting. To improve the performance of the rendering we introduced several improvements to one of the underlying building blocks, the A-Buffer. The method has been carefully analyzed with respect to memory consumption and performance. Its efficiency has been demonstrated on two different datasets.

FUTURE WORK. As discussed in Section 5 for future work, we see room for improvement with the sorting algorithm in the second pass of the A-Buffer. Besides that, the optimal resolution of the noise input, in comparison to the output resolution, could be studied in more detail, in order to improve performance while still keeping a good visual representation. Our current implementation is restricted to one volume data set. For the future we plan to support multiple volumes which should be a straightforward extension.

7. Acknowledgments

This work was supported in part by the Swedish e-Science Research Center (SeRC) and the Excellence Center at Linköping and Lund in Information Technology (ELLIIT). The described concepts have been realized using the Inviwo visualization framework (www.inviwo.org).

References

- [BK11] BARTA P., KOVÁCS B.: Order Independent Transparency with Per-Pixel Linked Lists. *The 15th Central European Seminar* on Computer Graphics (2011). 2
- [BM08] BAVOIL L., MYERS K.: Order independent transparency with dual depth peeling. NVIDIA OpenGL SDK (2008), 1–12. 2
- [BPE*15] BUSTAMANTE M., PETERSSON S., ERIKSSON J., ALEHAGEN U., DYVERFELDT P., CARLHÃĎLL C., EBBERS T.: Atlas-based analysis of 4d flow cmr: Automated vessel segmentation and flow quantification. Journal of Cardiovascular Magnetic Resonance (2015). 5
- [BPL*07] BAVOIL L., P.CALLAHAN S., LEFOHNM A., AO L. D. COMBA J., SILVA C. T.: Multi-fragment effects on the gpu using the k-buffer. *I3D '07 Proceedings of the 2007* symposium on Interactive 3D graphics and games. Pages 97-104 (2007). 2
- [CCF94] CABRAL B., CAM N., FORAN J.: Accelerated volume rendering and tomographic reconstruction using texture mapping hardware. *Proceedings of the 1994 Symposium on Volume Visualization* (1994), 91–98. 2
- [CL93] CABRAL B., LEEDOM L. C.: Imaging vector fields using line integral convolution. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques -SIGGRAPH '93* (1993), ACM, ACM Press, pp. 263–270. 2, 3
- [Cra10] CRASSIN C.: Opengl 4.0+ abuffer v2.0: Linked lists of fragment pages, 2010. 4, 5
- [EHK*06] ENGEL K., HADWIGER M., KNISS J., REZK-SALAMA C., WEISKOPF D.: *Real-Time Volume Graphics*. A K Peters, Ltd., Wellesley, Massachusetts, 2006. 2, 3
- [Eve01] EVERITT C.: Interactive order-independent transparency. NVIDIA OpenGL Applications Engineering 2, 6 (2001), 7. 2
- [FW08] FALK M., WEISKOPF D.: Output-sensitive 3D line integral convolution. *IEEE Transactions on Visualization and Computer Graphics* 14, 4 (2008), 820–834. 2, 3
- [Int97] INTERRANTE V.: Illustrating surface shape in volume data via principal direction-driven 3d line integral convolution. SIG-GRAPH '97 Proceedings of the 24th annual conference on Computer graphics and interactive techniques. Pages 109-116 (1997). 2
- [KM05] KAUFMAN A., MUELLER K.: Overview of volume rendering. Visualization Handbook d (2005), 127–174. 2, 3
- [KW03] KRUGER J., WESTERMANN R.: Acceleration techniques for GPU-based volume rendering. *Proceedings IEEE Vi*sualization (2003), 287–292. 2
- [LFS*14] LINDHOLM S., FALK M., SUNDÉN E., BOCK A., YNNERMAN A., ROPINSKI T.: Hybrid data visualization based on depth complexity histogram analysis. *Computer Graphics Forum* (2014). 2
- [LHD*04] LARAMEE R. S., HAUSER H., DOLEISCH H., VROLIJK B., POST F. H., WEISKOPF D.: The State of the Art in Flow Visualization: Dense and Texture-Based Techniques. *Computer Graphics Forum 23*, 2 (jun 2004), 203–221. 2
- [MCTB12] MAULE M., COMBA J. L. D., TORCHELSEN R., BASTOS R.: Memory-Efficient Order-Independent Transparency with Dynamic Fragment Buffer. 2012 25th SIBGRAPI Conference on Graphics, Patterns and Images (aug 2012), 134–141. 2
- [MHDG11] MUIGG P., HADWIGER M., DOLEISCH H., GRÖLLER E.: Interactive volume visualization of general polyhedral grids. *IEEE transactions on visualization and computer* graphics 17, 12 (dec 2011), 2115–24. 2

- [PDB] PDB: Structure of native and apo carbonic anhydrase ii and structure of some of its anion-ligand complexes. 6
- [RGW*03] ROETTGER S., GUTHE S., WEISKOPF D., ERTL T., STRASSER W.: Smart Hardware-Accelerated Volume Rendering. Symposium on Visualization (VISSYM '03) (2003), 231–238.
- [RSHTE99] REZK-SALAMA C., HASTREITER P., TEITZEL C., ERTL T.: Interactive exploration of volume line integral convolution based on 3D-texture mapping. *Proceedings Visualization* '99 (Cat. No.99CB37067) Li, section 6 (1999), 233–528. 2
- [SBF15] SCHOLLMEYER A., BABANIN A., FROEHLICH B.: Order-independent transparency for programmable deferred shading pipelines. In *Computer Graphics Forum* (2015), vol. 34, Wiley Online Library, pp. 67–76. 2
- [SEA08] SINTORN E., EISEMANN E., ASSARSSON U.: Sample based visibility for soft shadows using alias-free shadow maps. *Computer Graphics Forum* 27, 4 (2008), 1285–1292. 2
- [SH95] STALLING D., HEGE H.-C.: Fast and resolution independent line integral convolution. Proceedings of the 22nd annual conference on Computer graphics and interactive techniques -SIGGRAPH '95 (1995), 249–256. 2
- [SSK*15] SUNDÉN E., STENETEG P., KOTTRAVEL S., JÖNS-SON D., ENGLUND R., FALK M., ROPINSKI T.: Inviwo - An Extensible, Multi-Purpose Visualization Framework. Poster at IEEE Vis, 2015. 5
- [SSKE05] STEGMAIER S., STRENGERT M., KLEIN T., ERTL T.: A simple and flexible volume rendering framework for graphicshardware-based raycasting. *Fourth International Workshop on Volume Graphics*, 2005. (2005), 187–241. 2
- [SSKLK13] SHREINER D., SELLERS G., KESSENICH J., LICEA-KANE B.: OpenGL programming guide: The Official guide to learning OpenGL, version 4.3. Addison-Wesley, 2013. 3
- [vW03] VAN WIJK J.: Image based flow visualization for curved surfaces. IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control (2003), 123–130. 2
- [Zha14] ZHANG N.: Memory-hazard-aware k-buffer algorithm for order-independent transparency rendering. *IEEE transactions* on visualization and computer graphics 20, 2 (feb 2014), 238–48.