HARVEST4D

HARVESTING DYNAMIC 3D WORLDS FROM COMMODITY SENSOR CLOUDS

Publications for Task 8.2

Deliverable 8.21

Date:	9.7.2015
Grant Agreement number:	EU 323567
Project acronym:	HARVEST4D
Project title:	Harvesting Dynamic 3D Worlds from Commodity Sensor Clouds

Document Information

Deliverable number	D8.21
Deliverable name	Publications for Task 8.2
Version	0.4
Date	2015-07-09
WP Number	8
Lead Beneficiary	DELFT
Nature	R
Dissemination level	PU
Status	Final
Author(s)	DELFT

Revision History

Rev.	Date	Author	Org.	Description
0.1	2015-06-30	Timothy Kol	DELFT	Draft
0.2	2015-07-07	Timothy Kol	DELFT	Final
0.3	2015-07-08	Timothy Kol	DELFT	More final
0.4	2015-07-09	Timothy Kol	DELFT	Ultimate

Statement of originality

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

TABLE OF CONTENTS

1 EXECUTIVE SUMMARY

1.1 INTRODUCTION

This deliverable describes the publications that resulted from Task 8.2, and how they fit into the work plan of the project.

The objective of Task 8.2 is to visualize space-time data. This plays a crucial part in the Harvest4D project, because a user should be able to quickly interpret the content on the screen and browse information along the time axis. Special care has to be given to the temporal aspect. Standard downsampling is not an acceptable solution to give a good data overview because important changes might be missed. We provide specialized visualization strategies, using various cues in this context. Similarly to a video summary, we want to group significant changes, with respect to saliency, uncertainty, or geometry. Further, the visualization is influenced by external factors. Considering the user leads to more intuitive ways of accessing the data.

There are so far five publications that are mainly associated with Task 8.2. These can be found in the appendix of this deliverable. The publications are:

- Renata Raidou, Martin Eisemann, Marcel Breeuwer, Elmar Eisemann, Anna Vilanova
  Orientation-Enhanced Parallel Coordinate Plots (OPCPs) **(conditionally accepted)**
  IEEE Transactions on Visualization and Computer Graphics Special Issue, January 2016
- Johannes G. Leskens, Christian Kehl, Tim Tutenel, Timothy R. Kol, Gerwin de Haan, Guus S. Stelling, Elmar Eisemann
  An interactive simulation and visualization tool for flood analysis usable for practitioners
  Mitigation and Adaptation Strategies for Global Change p 1-18, 2015
- Tim Tutenel, Christian Kehl, Elmar Eisemann
  Interactive visual analysis of flood scenarios using large-scale LiDAR point clouds
  Geospatial World Forum, 2013
- Leila Schemali, Elmar Eisemann
  ChromoStereoscopic Rendering for Trichromatic Displays
  Proceedings of Expressive (NPAR) p 57-62, 2014
- Daniel van der Ende, Jean-Marc Thiery, Elmar Eisemann
  Accelerated Mean Shift For Static And Streaming Environments
  Proceedings of Data Analytics, 2015

Two other papers are related to Task 8.2, found in the deliverables they mainly contribute to:

- Paolo Cignoni, Nico Pietroni, Luigi Malomo, Roberto Scopigno
  Field-Aligned Mesh Joinery (Task 8.1)
  ACM Transactions on Graphics (TOG), Volume 33 (1), 2014

- Leila Schemali, Elmar Eisemann
  Design and Evaluation of Mouse Cursors in a Stereoscopic Desktop Environment (Task 8.4)
  IEEE 3DUI Techical Notes, 2014

2 DESCRIPTION OF PUBLICATIONS

2.1 OVERVIEW

One of the challenges of space-time data is that you are interacting with 4D data. Here, classical 2D and 3D representations fall short. For this we developed a novel technique [Raidou et al. 2016] based on parallel coordinate plots, with which we can both visualize and interact with higher dimensional data. Another approach to visualize space-time data is of course to decouple the time dimension, and instead animate the 3D data over time to provide the user with a more intuitive visualization method. An application of this is for instance the visualization of flood simulations to aid practitioners (policy analysts and political decision-makers), a topic that fits well in the Harvest4D project. Our system allows the user to interactively control the simulation process (addition of water sources or influence of rainfall), while a realistic visualization allows the user to mentally map the results onto the real world. One challenge in this lies in the fact that the dataset consists of a high-resolution point set of a complete country, numbering in multiple terabytes of information. To handle this, we make use of a hierarchical data structure, only streaming into memory the parts that are visible [Tutenel et al. 2013]. While 3D data can be animated to visualize 4D information, we are still projecting this 3D data on a 2D screen. This can at times make it difficult to perceive the third dimension (depth), potentially providing the user with an insufficient understanding of the data. To alleviate this problem, much research has focused on simulating the human 3D vision on a 2D screen. We proposed a novel method based on chromo-stereoscopy, using the ChromaDepth glasses to provide users with an improved depth perception on regular trichromatic displays, while the image remains backwards compatible; i.e. users without the glasses still see a crisp and clear image [Schemali and Eisemann 2014]. Finally, for many visualization applications, clustering high dimensional data is an important aspect, e.g. to remove clutter or highlight patterns. For this, the mean shift technique is a popular algorithm, thanks to its ability to find non-convex and local clusters. However, it suffers from relatively poor computational performance. We have devised an acceleration scheme for the mean shift algorithm [Van der Ende et al. 2015], which handles both static and streaming environment, making it an invaluable tool for space-time data visualization.

2.2 ORIENTATION-ENHANCED PARALLEL COORDINATE PLOTS (OPCPS)

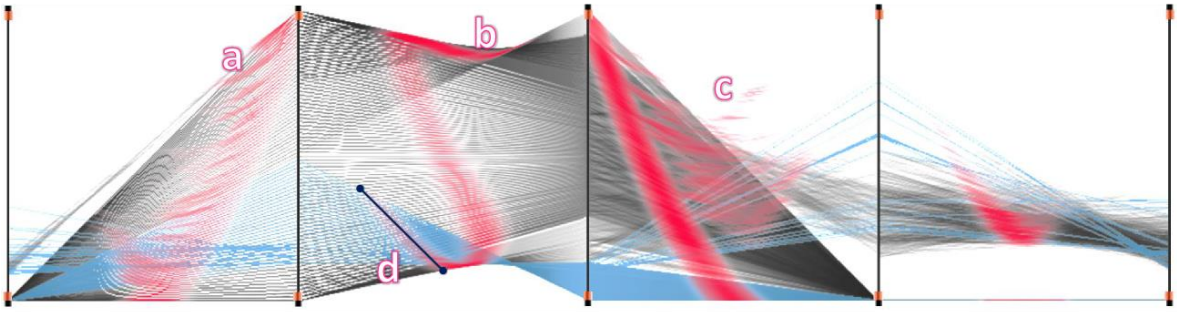


Figure 1: The proposed OPCPs (red), applied to the Venus dataset: (a) Visual enhancement of small patterns between the first two dimensions of the data, i.e., small structures obstructed by a strong pattern. - (b) Facilitated identification of distinct patterns between the second and third data dimension. - (c) Improved readability of outliers, i.e., low density information areas, in the representation. - (d) Efficient and accurate selection (blue) of a specific data structure, using the proposed O-Brushing (dark blue line).

In this paper, we introduce a new visualization method [Raidou et al. 2016] based on parallel coordinate plots, one of the most powerful techniques for the visualization of multivariate data. However, for large datasets, the representation suffers from clutter due to overplotting. In this case, discerning the underlying data information and selecting specific interesting patterns can become difficult. We propose a new and simple technique to improve the display of PCPs by emphasizing the underlying data structure. Our Orientation-enhanced Parallel Coordinate Plots (OPCPs) improve pattern and outlier discernibility by visually enhancing parts of each PCP polyline with respect to its slope. This enhancement also allows us to introduce a novel and efficient selection method, the Orientation-enhanced Brushing (O-Brushing). Our solution is particularly useful when multiple patterns are present or when the view on certain patterns is obstructed by noise. In the paper, we present the results of our approach with several synthetic and real-world datasets. Also, we conducted a user evaluation, which verifies the advantages of the OPCPs in terms of discernibility of information in complex data. It also confirms that O-Brushing eases the selection of data patterns in PCPs and reduces the amount of necessary user interactions compared to state-of-the-art brushing techniques. A visualization of the 5D Venus dataset using our OPCP framework is shown in Figure 1.

2.3 AN INTERACTIVE SIMULATION AND VISUALIZATION TOOL FOR FLOOD ANALYSIS FOR PRACTITIONERS

In this paper, we present tool for visualizing flood simulations to enable easier and more accessible flood analysis for non-experts, such as policy analysts and decision-makers [Leskens et al. 2015]. After all, developing strategies to mitigate or to adapt to the threats of floods is an important topic in the context of climate changes. Many of the world's cities are endangered due to rising ocean levels and changing precipitation patterns. It is therefore crucial to develop analytical tools that allow us to evaluate the threats of floods and to investigate the influence of

mitigation and adaptation measures, such as stronger dikes, adaptive spatial planning, and flood disaster plans. Up until the present, analytical tools have only been accessible to domain experts, as the involved simulation processes are complex and rely on computational and data-intensive models. Outputs of these analytical tools are presented to practitioners (i.e., policy analysts and political decision-makers) on maps or in graphical user interfaces. In practice, this output is only used in limited measure because practitioners often have different information requirements or do not trust the direct outcome.

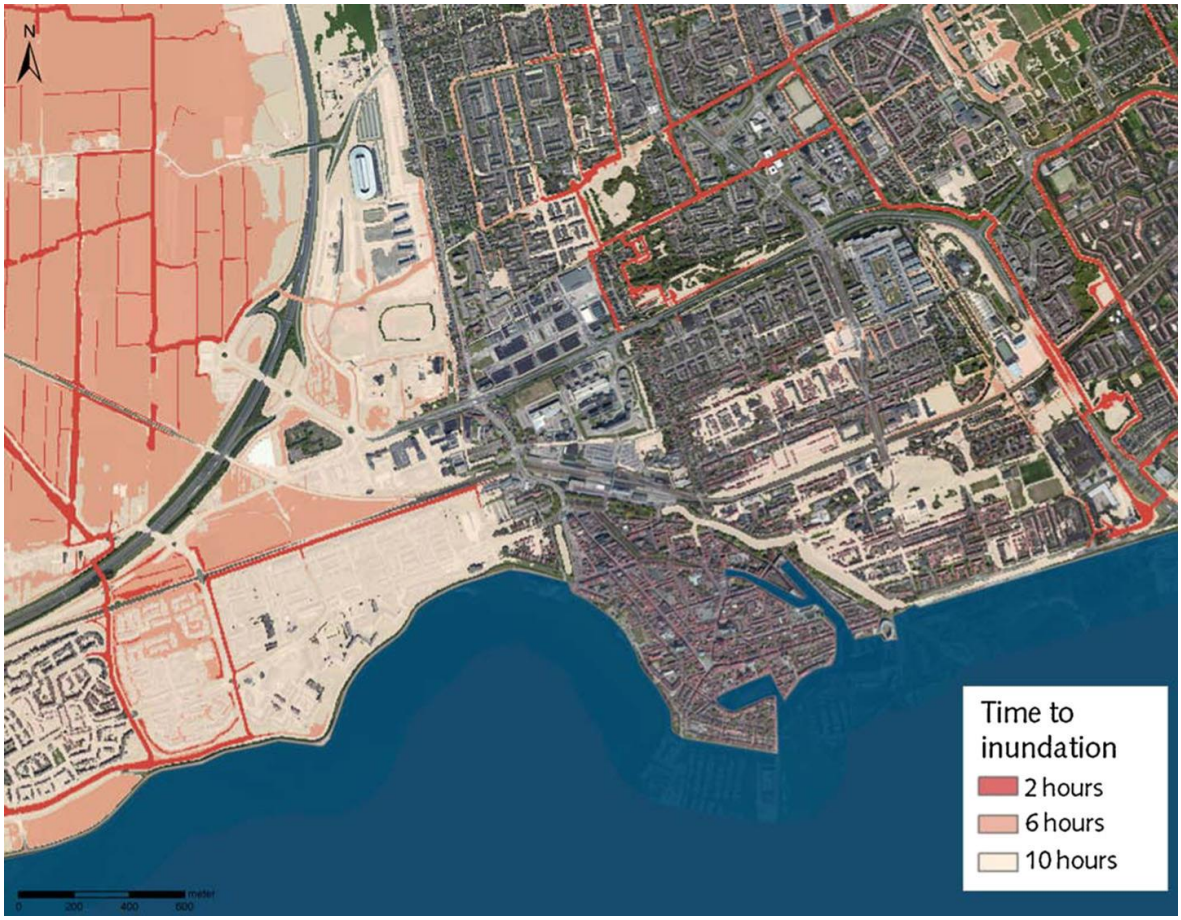


Figure 2: An exemplary map showing flood arrival times in different zones, which can be used for city evacuation planning.

Nonetheless, literature indicates that a closer collaboration between domain experts and practitioners can ensure that the information requirements of practitioners are better aligned with the opportunities and limitations of analytical tools. The objective of our work is to present a step forward in the effort to make analytical tools in flood management accessible for practitioners to support this collaboration between domain experts and practitioners. Our system allows the user to interactively control the simulation process (addition of water sources or influence of rainfall), while a realistic visualization allows the user to mentally map the results

onto the real world (see e.g. Figure 2). We have developed several novel algorithms to present and interact with flood data. We explain the technologies, discuss their necessity alongside test cases, and introduce a user study to analyze the reactions of practitioners to our system. We conclude that, despite the complexity of flood simulation models and the size of the involved data sets, our system is accessible for practitioners of flood management so that they can carry out flood simulations together with domain experts in interactive work sessions. Therefore, this work has the potential to significantly change the decision-making process and may become an important asset in choosing sustainable flood mitigations and adaptation strategies.

2.4 INTERACTIVE VISUAL ANALYSIS OF FLOOD SCENARIOS USING LARGE-SCALE LIDAR POINT CLOUDS

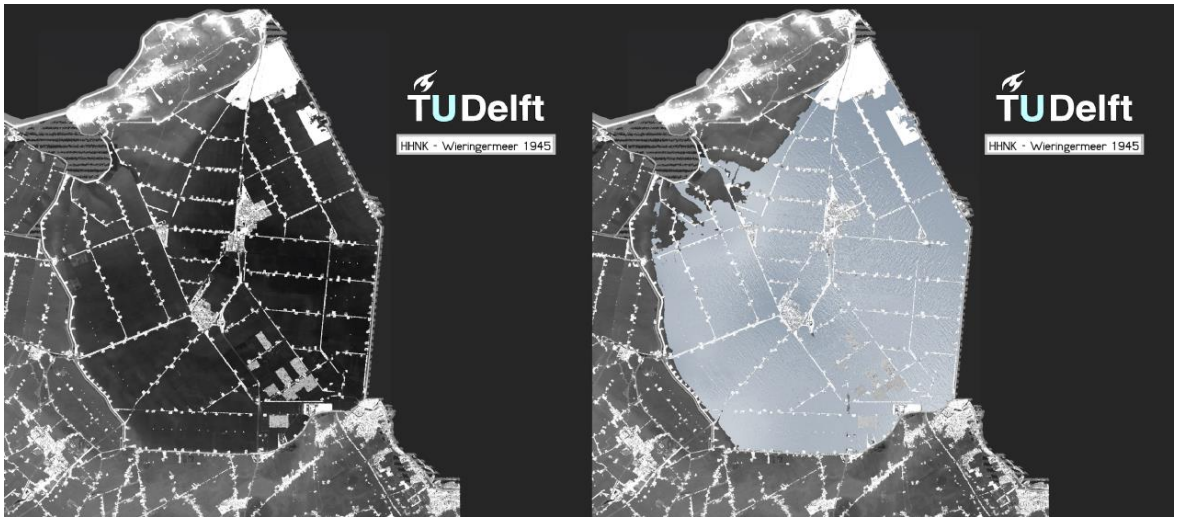


Figure 3: By visually analyzing the result of the calibrated flood simulation, following facts are to note: water distributes quickly along channels, so that adjacent areas are quickly inundated; in the low plain rescuing higher grounds are scarce; and the polder acts as a basin.

Of course, the visualization of large-scale geospatial data, that is required for the tool described in the previous section, is a demanding challenge. In this work [Tutenel et al. 2013], we propose a novel solution that is able to visualize colored LiDAR data of several hundred square kilometers including high-resolution flood simulation results. One key contribution is a level-of-detail rendering algorithm that enables us to handle terabytes of data interactively. Furthermore, we present a novel algorithm to annotate (color entire regions or highlight routes), or modify (i.e. clip, cut, raise/lower) the unordered LiDAR point set on the fly with flexible 2D geographic metadata (i.e., polygons, paths and landmarks). The principles of our technique are to make use of hierarchical structures on the metadata and a new quadtree-based GPU traversal algorithm. We examined three test cases and show that combining the flood simulation with different topographic shading techniques facilitates drawing conclusions about water flow, security measures, and evacuation planning, as showcased by Figure 3.

2.5 CHROMOSTEREOSCOPIC RENDERING FOR TRICHROMATIC DISPLAYS

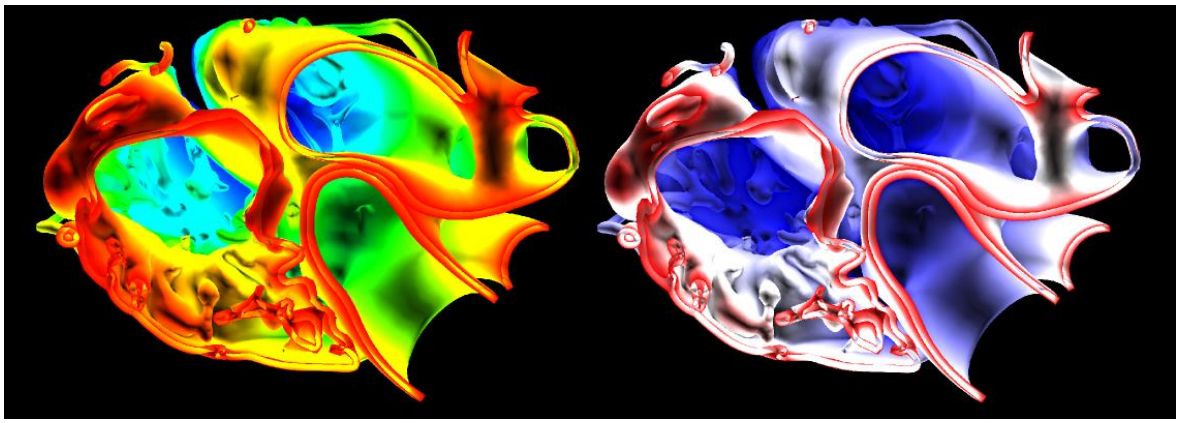


Figure 4: ChromaDepth glasses act like a prism that disperses incoming light and induces a differing depth perception for different light wavelengths. As most displays are limited to mixing three primaries (RGB), the depth effect can be significantly reduced, when using the usual mapping of depth to hue (left). Our red to white to blue mapping and shading cues achieve a significant improvement (right).

In the aforementioned works, we animate 3D data to visualize what is effectively 4D information. However, since we are still showing this 3D information on a 2D screen, it can be difficult to perceive depth, limiting the user's understanding of the scene. In this paper, we were inspired by the chromostereopsis phenomenom to develop a chromostereoscopic rendering method to improve depth perception [Schemali and Eisemann 2014]. The chromostereopsis phenomenom leads to a differing depth perception of different color hues, e.g., red is perceived slightly in front of blue. In chromostereoscopic rendering, 2D images are produced that encode depth in color. While the natural chromostereopsis of our human visual system is rather low, it can be enhanced via ChromaDepth glasses, which induce chromatic aberrations in one eye by refracting light of different wavelengths differently, hereby offsetting the projected position slightly in one eye. Although, it might seem natural to map depth linearly to hue, which was also the basis of previous solutions, we demonstrate that such a mapping reduces the stereoscopic effect when using standard trichromatic displays or printing systems (see Figure 4). We propose an algorithm, which enables an improved stereoscopic experience with reduced artifacts.

2.6 ACCELERATED MEAN SHIFT FOR STATIC AND STREAMING ENVIRONMENTS

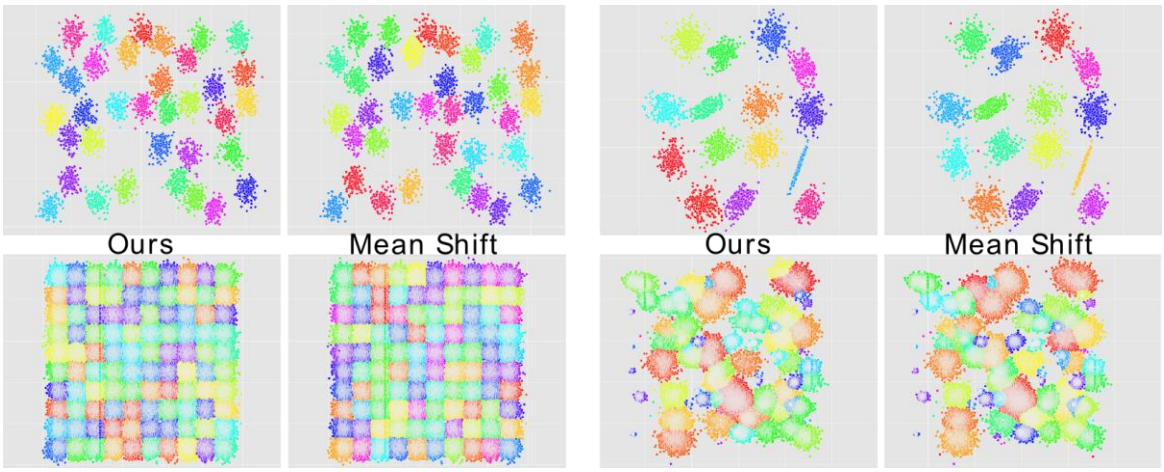


Figure 5: Comparison with mean shift on 2D data.

In this work, we present an acceleration method for the mean shift algorithm [Van der Ende et al. 2015]. This well-known clustering algorithm has attractive properties such as the ability to find non-convex and local clusters even in high dimensional spaces, while remaining relatively insensitive to outliers. However, due to its poor computational performance, real-world applications are limited. We propose a novel acceleration strategy for the traditional mean shift algorithm, along with a two-layer strategy, resulting in a considerable performance increase, while maintaining high cluster quality. We also show how to find clusters in a streaming environment with bounded memory, in which queries can be answered at interactive rates, and for which no mean shift-based algorithm currently exists. Our online structure can be updated at very minimal cost and as infrequently as possible, and we show how to detect the time at which this update needs to be triggered. Our technique is validated extensively in both static and streaming environments (see Figure 5).

3 REFERENCES

- Renata Raidou, Martin Eisemann, Marcel Breeuwer, Elmar Eisemann, Anna Vilanova
  Orientation-Enhanced Parallel Coordinate Plots (OPCPs) **(conditionally accepted)**
  IEEE Transactions on Visualization and Computer Graphics Special Issue, January 2016
- Johannes G. Leskens, Christian Kehl, Tim Tutenel, Timothy R. Kol, Gerwin de Haan, Guus S. Stelling, Elmar Eisemann
  An interactive simulation and visualization tool for flood analysis usable for practitioners
  Mitigation and Adaptation Strategies for Global Change p 1-18, 2015

- Tim Tutenel, Christian Kehl, Elmar Eisemann
  Interactive visual analysis of flood scenarios using large-scale LiDAR point clouds
  Geospatial World Forum, 2013
- Leila Schemali, Elmar Eisemann
  ChromoStereoscopic Rendering for Trichromatic Displays
  Proceedings of Expressive (NPAR) p 57-62, 2014
- Daniel van der Ende, Jean-Marc Thiery, Elmar Eisemann
  Accelerated Mean Shift For Static And Streaming Environments
  Proceedings of Data Analytics, 2015

4 APPENDIX

The following pages contain all the publications that are directly associated with this deliverable. Other publications referenced in this deliverable can be found in the public Harvest4D webpage (for already published papers), or in the restricted section of the webpage (for papers under submission, conditionally accepted papers, etc.).

Orientation-Enhanced Parallel Coordinate Plots

Renata Georgia Raidou, Martin Eisemann, Marcel Breeuwer, Elmar Eisemann, Anna Vilanova

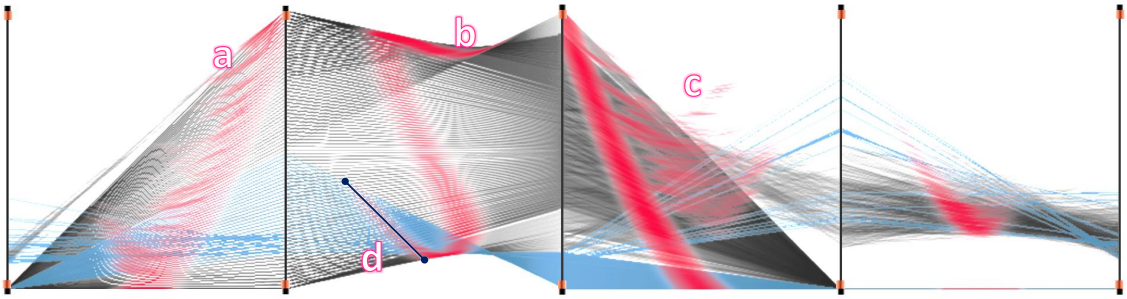


Fig. 1. The proposed OPCPs (red), applied to the `Venus` dataset [2]: (a) Visual enhancement of small patterns between the first two dimensions of the data, i.e., small structures obstructed by a strong pattern. - (b) Facilitated identification of distinct patterns between the second and third data dimension. - (c) Improved readability of outliers, i.e., low density information areas, in the representation. - (d) Efficient and accurate selection (blue) of a specific data structure, using the proposed O-Brushing (dark blue line).

Abstract—Parallel Coordinate Plots (PCPs) is one of the most powerful techniques for the visualization of multivariate data. However, for large datasets, the representation suffers from clutter due to overplotting. In this case, discerning the underlying data information and selecting specific interesting patterns can become difficult. We propose a new and simple technique to improve the display of PCPs by emphasizing the underlying data structure. Our Orientation-enhanced Parallel Coordinate Plots (OPCPs) improve pattern and outlier discernibility by visually enhancing parts of each PCP polyline with respect to its slope. This enhancement also allows us to introduce a novel and efficient selection method, the Orientation-enhanced Brushing (O-Brushing). Our solution is particularly useful when multiple patterns are present or when the view on certain patterns is obstructed by noise. We present the results of our approach with several synthetic and real-world datasets. Finally, we conducted a user evaluation, which verifies the advantages of the OPCPs in terms of discernibility of information in complex data. It also confirms that O-Brushing eases the selection of data patterns in PCPs and reduces the amount of necessary user interactions compared to state-of-the-art brushing techniques.

Index Terms—Parallel Coordinates, Orientation-enhanced Parallel Coordinates, Brushing, Orientation-enhanced Brushing, Data Readability, Data Selection

1 Introduction

Parallel Coordinate Plots (PCPs) [20] are widely used for the visualization of multivariate data. Here, the multiple dimensions of the data are mapped one-by-one to a number of parallel vertical axes. Each multidimensional object in the dataset is mapped to a polyline that intersects the axes, connecting the scalar values of every dimension [20]. PCPs efficiently display in a single view all 2D projections of adjacent data dimensions [19, 21, 39], enabling the identification of relations among dimensions and the detection of patterns or trends in the data - especially with the help of interaction [17, 34], such as brushing [16] or reordering [4, 30, 36].

A limitation of PCPs is that they might suffer from clutter, due to overplotting [17]. This causes problems in the exploration and interpretation of the data, especially in high density data parts. Reducing visual clutter in PCPs is an important topic [5, 11, 27, 29, 45]. However, most of the previous solutions are complex and focus mainly on aiding the detection of clusters in the data [5, 45], not in revealing the overall data structure. In other cases, the proposed visualizations may even unintentionally lead to concealing patterns and outliers - or to making them less discernible [5, 29]. Finally, some solutions require interaction to achieve clutter reduction [11, 27].

We propose a new and simple technique to improve the representation of datasets in PCPs: the *Orientation-enhanced Parallel Coordinates* (OPCPs). Our technique visually enhances specific parts of each PCP line, depending on its slope. Hereby, it enables discerning individual trends and patterns, while it may even reveal patterns that are potentially obscured in traditional PCPs. This enhancement also allows us to introduce a new brushing technique, the *Orientation-enhanced Brushing* (O-Brushing), to facilitate pattern selection in complex data.

Our paper presents the following two major *contributions*:

- The concept of Orientation-enhanced Parallel Coordinates (OPCPs) to improve the view and discernibility of patterns in otherwise cluttered PCPs, without loss of low density data information or outliers.
- A versatile brushing technique based on the OPCPs: the Orientation-enhanced Brushing (O-Brushing). It enables efficient selection of individual data structures, with reduced user interaction.

2 Related Work

Many different techniques have been proposed for the enhancement of data display and clutter reduction in Information Visualization representations [9], including PCPs. Some approaches require the manipulation of the axes of the representation, using *reordering* [4, 30, 36, 42]. These approaches are able to reveal hidden patterns and facilitate data interpretation. However, in data with a large number of points reordering is insufficient. Other approaches involve *visual enhancement* of PCPs by rendering curves instead of lines [3, 14, 35, 45]. Such approaches are especially effective in reducing clutter at the crossings of PCP lines, but they might suppress data patterns, such as outliers.

Another commonly encountered group of techniques requires *clustering*, combined with different kinds of visual enhancements, such as: manipulating PCPs by averaging polylines and visualizing correlation coefficients between polyline subsets [33]; filtering PCPs based

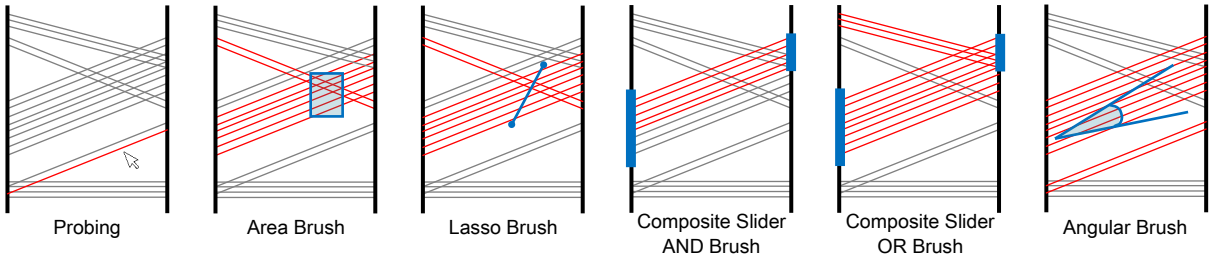


Fig. 2. Overview of different state-of-the-art brushing approaches for PCPs. With red we denote the selections in each case, with blue we denote the brushing operation.

on frequency or density of the data [5]; combining polyline splatting for cluster detection and segment splatting for clutter reduction [44]; using cluster-based hierarchical enhancements and proximity-based coloring schemes to provide a multiresolution view to the data [11]; enabling context visualization at several levels of abstraction, both for the representation of outliers and trends [29]; or using several transfer functions to reveal specific clusters and patterns in the data [22]. All previously mentioned cases involve clustering methods and they focus on detecting and differentiating specific clusters or trends in the data - not data patterns or underlying structures. In certain cases, clustering solutions inevitably lose information in low density areas, when reducing overplotting in high density areas. The approach of Zhou et al. [44] even requires animation, which is not always feasible. Finally, a more artistic approach was proposed by McDonnell et al. [27]. It incorporates a variety of techniques when rendering PCPs, such as edge-bundling, visualization of the distribution and density of the data via opacity and shading, or silhouettes for easy distinction of overlapping clusters. However, not all of these techniques can be used in a single view as some of them do not work well if combined.

PCPs have also been used in *combination* with other representations, such as Star Glyphs [10], radviz [6], or scatterplots [43]. As recognized by Holten et al. [18], combining scatterplots with PCPs outperforms many other PCP variants, such as combining with colors, opacity, curved polylines or animations. PCPs have also been combined with histograms [12], to simultaneously show the density and slopes of polylines. This combination enables the exploration of clusters, linear correlations and outliers in large datasets, with more emphasis on data-driven and not pattern-revealing exploration.

Interaction makes local and dynamic data enhancements possible. The use of lenses [8, 40] or brushing are typical examples. As part of the XmdvTool [37], a number of different brushes have been proposed by Martin et al. [26] and Ward [38]. Depending on the information that needs to be shown in the data, different brushes are used for highlighting, linking or masking the underlying data. Additionally, wavelet approximations are used to enhance brushing [41], by showing different parts of the polylines at different resolutions. However, brushing two variables in a non-separable way has only been enabled by the angular brushing proposed by Hauser et al. [16]. The most important state-of-the-art brushing approaches are presented in Fig. 2.

In summary, there are different approaches for data enhancement and readability improvement in PCPs. However, most of the solutions aim at reducing clutter in PCPs, by clustering the data, without giving a better understanding of the overall underlying structure. Data details, such as outliers, are often unintentionally hidden. Additionally, some solutions work better - or only - on a screen, either because they are animated or because they require interaction. Finally, most of the approaches require complex steps, which means that they cannot always be easily reproduced or used. In the following sections, we present our approach to handle these challenges.

3 Orientation-Enhanced Approach for PCPs

Our solution consists of two main components: the Orientation-enhanced Parallel Coordinate Plots (OPCPs) for visual enhancement of PCPs (Sec. 3.2), and the Orientation-enhanced Brushing (O-Brushing) for interactive selection and analysis in OPCPs (Sec. 3.3).

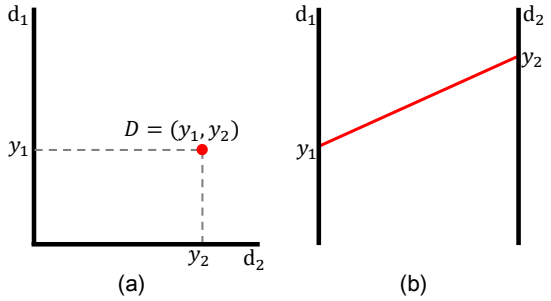


Fig. 3. Schematic representation of the concept of PCPs for the simple case of a two-dimensional point D with dimensions d_1 and d_2 and dimension values (y_1, y_2): (a) In a scatterplot. - (b) In a PCP.

3.1 Background: Parallel Coordinate Plots

In a simple two-dimensional dataset with dimensions d_1 and d_2, a data point $D = (y_1, y_2)$ is plotted as a PCP line, intersecting the two vertical axes d_1 and d_2 at the positions y_1 and y_2, respectively (Fig. 3). When plotting the PCP (poly)lines, it is common to employ opacity, as a simple way of representing the density of the lines [17]. We refer to this enhancement as *Density PCPs*, from now on.

3.2 Orientation-enhanced Parallel Coordinate Plots

Holten et al. [18] conducted an evaluation of PCP variants, where they demonstrated that no other enhancement from the examined alternatives improves PCPs significantly, apart from combining scatterplots with PCPs. Inspired by this paper, we investigated a simple way to combine effectively the two representations to enhance the display of PCPs. A similar approach was followed by Yuan et al. [43]. However, the latter is complex and requires bending the polylines to fit to the points of the scatterplots. In contrast, we are looking for an approach that is simple and should not require polyline bending, hereby keeping the original appearance of PCPs intact.

The goal of the proposed visual enhancement of PCPs is to provide a better understanding in the visualized data, by integrating PCPs and their corresponding scatterplots. In many papers, the combination of PCPs with scatterplots has been limited to having multiple interactive linked views. This might entail memory limitations, as a result of switching between the two separate representations. In the proposed OPCPs, the basic principle is to enhance the PCP lines with respect to their slope. This solution links PCPs and the corresponding scatterplot of the neighboring two axes in a natural way.

Mapping. For illustration purposes, we demonstrate our concept using a two-dimensional case. In the following example, we assume for simplicity that the data values for each dimension have been normalized to the range $[0,1]$. A PCP line, as shown in Fig. 4-(a), is defined by its dimension values (y_1, y_2) and a slope α:

$$\alpha = \frac{y_2 - y_1}{d_x}, \tag{1}$$

where d_x is the distance between the two vertical PCP axes.

We map this PCP line to a unique reference point $P = (x_p, y_p)$ in the space between the two PCP axes, with $x_p \in [0, d_x]$ and $y_p \in [0, d_y]$, where d_y is the length of the vertical axis (Fig. 4-(a)). The slope in

Eq.(1) is linearly mapped to x_p, while y_p is chosen to make P lie on its corresponding PCP line:

$$x_p = \frac{d_x^2}{2d_y} \cdot \alpha + \frac{d_x}{2} \quad (2)$$

$$y_p = y_1 + x_p \cdot \alpha \quad (3)$$

In Fig. 4-(b), we show an example with multiple PCP lines and their respective reference points. Eqs.(2,3) result into a point-to-point transformation, i.e. a warping, of a 2D scatterplot space to the OPCP space, as shown in Fig. 5. This illustration shows the link between the scatterplot points and the reference point positions on the PCP lines.

Representation. To visually enhance each reference point and to preserve the orientation and context of its PCP line, we create a small line segment: the Orientation-enhanced PCP. It is a small segment that shares the original PCP line orientation and is centered at the reference point P. Assigning a constant intensity and a given length to each segment would result into OPCP segments that would not be visually separated, if they would be very close to each other. Therefore, we vary the intensity of the segments using a kernel smoother: we smooth the edges of the segments and assign higher intensities in the middle, i.e. at the reference point P (Fig. 6). This desired intensity profile can be achieved with peak-shape kernels, such as a Gaussian [13]. The intensity I at a point $S = (x_s, y_s)$ of the OPCP segment, resulting from a reference point $P = (x_p, y_p)$ after applying the Gaussian kernel, is described as:

$$I(x_s, y_s) = k \cdot \exp\left(- \frac{\|x_s - x_p\|^2}{2 \cdot \sigma^2} \right), \quad (4)$$

where σ is the bandwidth of the kernel, which is user-defined and k is a scale factor (height of the peak) given by $\frac{1}{\sigma\sqrt{2\pi}}$. The bandwidth controls the length of the OPCP segments: larger σ values result in smoother and wider-spread segments. Fig. 7 shows an example of OPCPs applied to three simple synthetic cases and the effect of σ on their appearance.

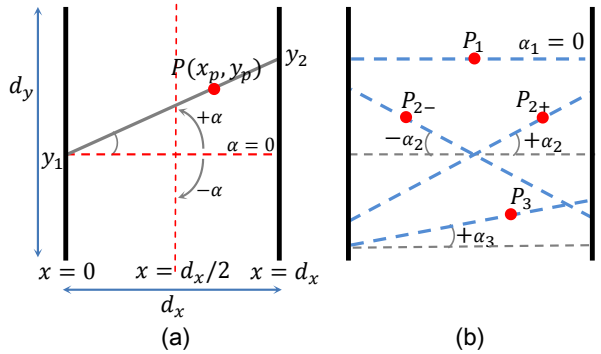


Fig. 4. Schematic representation of the concept behind OPCPs: (a) Mapping of the slope α of the PCP line (y_1, y_2) to the reference point $P = (x_p, y_p)$ between the two PCP axes. - (b) Mapping of the slopes of multiple PCP lines to their corresponding reference points.

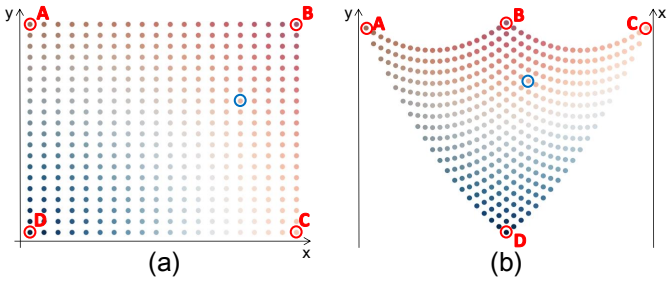


Fig. 5. Schematic representation of the transformation from (a) the scatterplot space to (b) the OPCP space, using a 2D colormap [32] to show the injective point-to-point correspondence.

Visual Enhancement. In the paper of Harrison et al. [15], it is stated that PCPs can emphasize specific correlations more than others. Depending on the data aspects that need to be emphasized, we propose

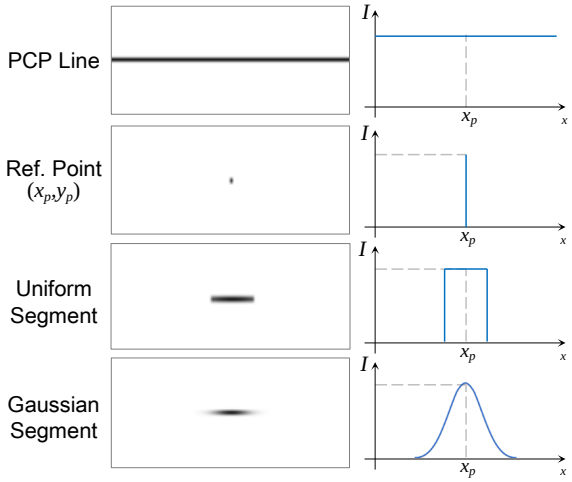


Fig. 6. Alternatives considered for the intensity encoding of the OPCP segments. Next to each case, we show also the intensity profile.

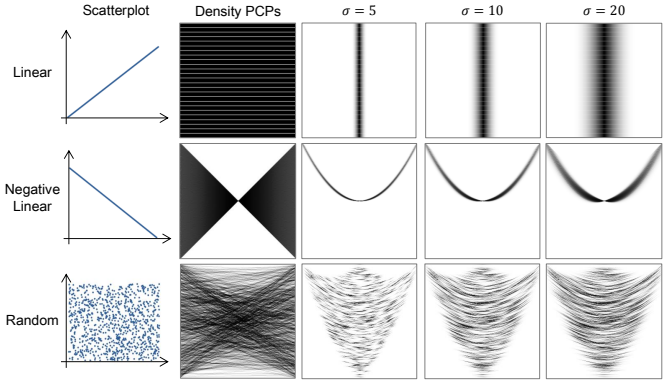


Fig. 7. Effect of the σ value of the Gaussian kernel on OPCPs, for three simple synthetic cases. To increase the visibility of the segments, we have linearly scaled the image intensities to the range [0,1].

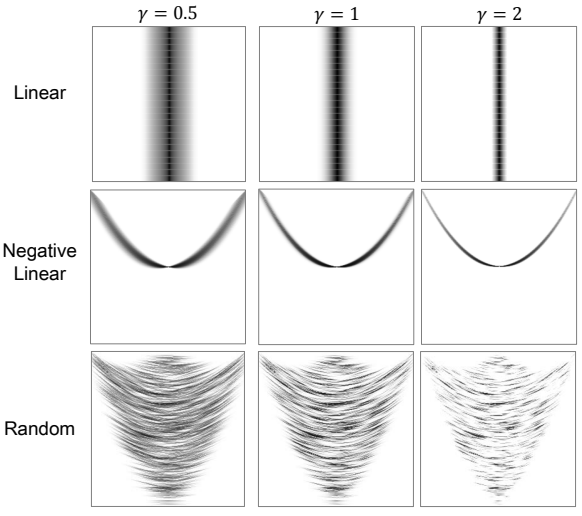


Fig. 8. Effect of the gamma correction on the appearance of the OPCPs. Here, the σ was set to 10 and the image intensities were scaled to the range [0,1].

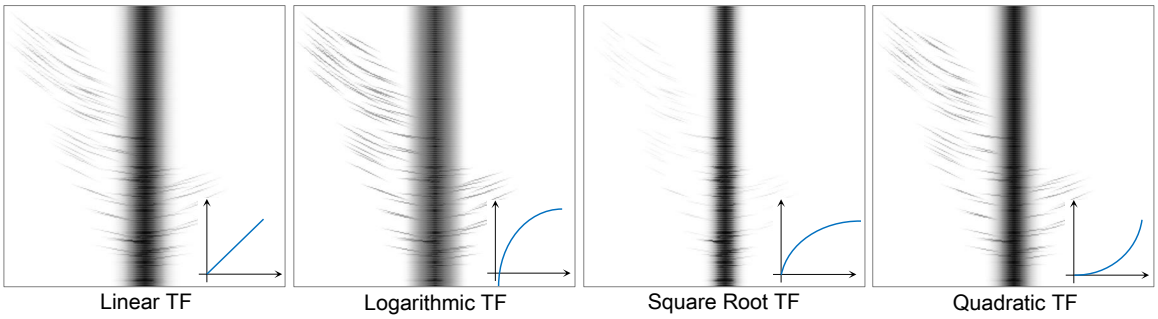


Fig. 9. Enhancement of different data aspects, using transfer functions (TFs). Here, we use a synthetic dataset with a dominant linear relation among the dimensions and a few outliers. The bandwidth σ was set to 10 and the image intensities were scaled to the range [0,1].

to optionally employ three enhancement methods: gamma correction, transfer functions, and histogram equalization.

Gamma correction [13] allows the user to remap the levels of the intensity range in order to discern more details in the darker parts of the OPCP segments. This can be accomplished with low values of gamma, while increasing values of gamma sharpen the OPCPs. Gamma correction is applied per pixel, transforming the intensity I to $I_{gcorr} = I^{\gamma}$. The effect of the parameter γ is depicted in Fig. 8.

The effect of gamma correction can be generalized by applying a transfer function (TF), aiming at controlling the contrast in the representation. As introduced in the work of Johansson et al. [22], different TFs affect the appearence of different data aspects: a linear TF gives an overview on the data, a logarithmic TF enhances low density areas, a square root TF emphasizes outliers in the data and a quadratic TF enhances the high density areas. The effect of the four previously mentioned TFs is shown in Fig. 9.

Optionally, histogram equalization [13] reassigns the intensity values of an image such that the output will exhibit a uniform distribution of intensities. Histogram equalization can create a background-foreground effect and enable better discernibility of different patterns in the data, especially in the presence of noise or of strong patterns. The impact of histogram equalization in OPCPs is depicted in Fig. 10.

Overlay. We enhance PCPs by overlaying the OPCP segments on top of the traditional PCP polylines, e.g., on Density PCPs, using alpha blending [31] (Fig. 11-(a)). Overlaying OPCPs on top of PCPs helps in preserving their main benefit, namely the connectivity across data dimensions. In this way, PCP polyline bundles can still be traced. Additional color encoding of OPCPs can enhance and visually separate them from the underlying PCPs (Fig. 11-(b)). To reduce as much as possible user distraction from overlaying OPCPs on the PCP polylines and interfering with PCP bundle tracking, the appearence of OPCPs can be adjusted. The user can modify the color and opacity of the OPCP segments, but also to fine-tune the σ and γ values to make the OPCPs more or less prominent. For the purpose of this paper, we decided to encode the OPCP intensities as black color values in the explanatory examples and red in the overlay examples.

Parameter values. The parameters involved in the visual enhancement of the OPCPs, such as the bandwidth σ and the gamma correction value γ, should depend on the specific aspects of the data that needs to be brought forward. Therefore, we do not assign a specific set of values, but leave them user-controllable. In our interactive tool, we initially assign a set of values (σ=10 and γ=1), which give already a good result, but can be changed adequately by the user.

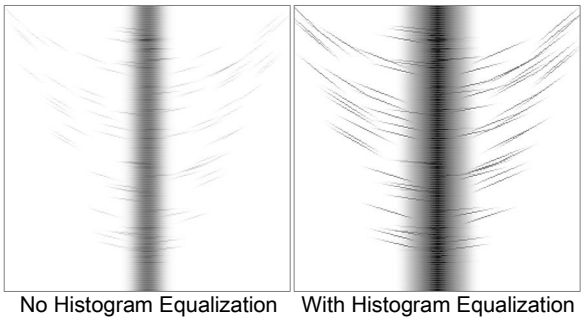


Fig. 10. Effect of histogram equalization on the appearance of OPCPs. Here, the σ was set to 10 and γ to 1.

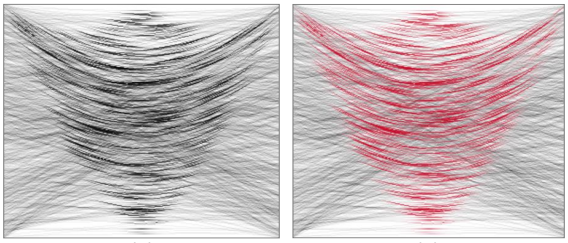


Fig. 11. Example of alpha blending in random noisy data: (a) Alpha blending of histogram equalized OPCPs with Density PCPs - (b) Color encoding of the foreground blended OPCPs (red) and background Density PCPs (black).

3.3 Orientation-enhanced Brush (O-Brushing)

Brushing [17, 34] is a common approach for data selection in PCPs. However, selection becomes more difficult when the amount of plotted lines increases. The previously proposed OPCPs establish for each 2D data point a unique position in the space between each pair of the PCP axes. This property allows us to introduce a new brushing approach that is applied in the OPCP space, for easier and more efficient selection of specific data patterns: the Orientation-enhanced Brushing (O-Brushing).

O-Brushing is performed on OPCP segments, in two ways: either with a traditional brush metaphor (O-Brush), or with a prober (O-Prober) (Fig. 12). The O-Brush (Fig. 12-(a)) acts as a lasso brush [17], applied only in the OPCP space and requires two user interactions, i.e., two clicks. The O-Prober (Fig. 12-(b)) is an interactive rectangle that can be resized and moved around the representation. It works simi-

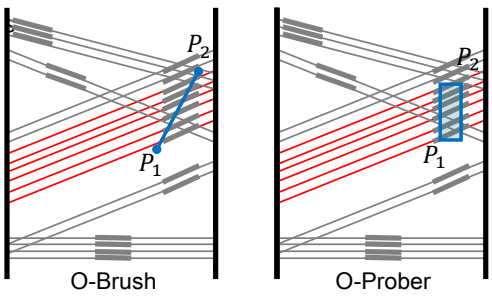


Fig. 12. Schematic representation of the concept behind O-Brushing. The thick gray segments represent OPCPs for each underlying PCP line. With red we denote the selections in each case, while with blue the brushing operation.

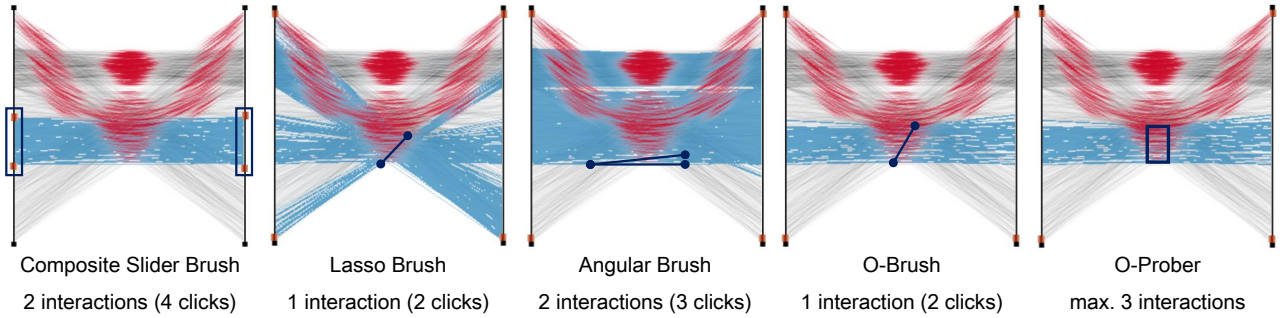


Fig. 13. Example for the comparison of O-Brush and O-Prober against traditional brushing methods, when attempting to select the same part, i.e., data points with middle values of both dimensions. All brushes have been applied individually to the data. In this example, the lasso and area brush do not succeed in selecting the specific data region. We show also the number of user interactions, i.e., the number of clicks, required for each of the brushes. The composite slider brush requires maximally four clicks, the O-Brush requires two clicks and the O-Prober requires maximally 3 user interactions (resize in both dimensions and translate).

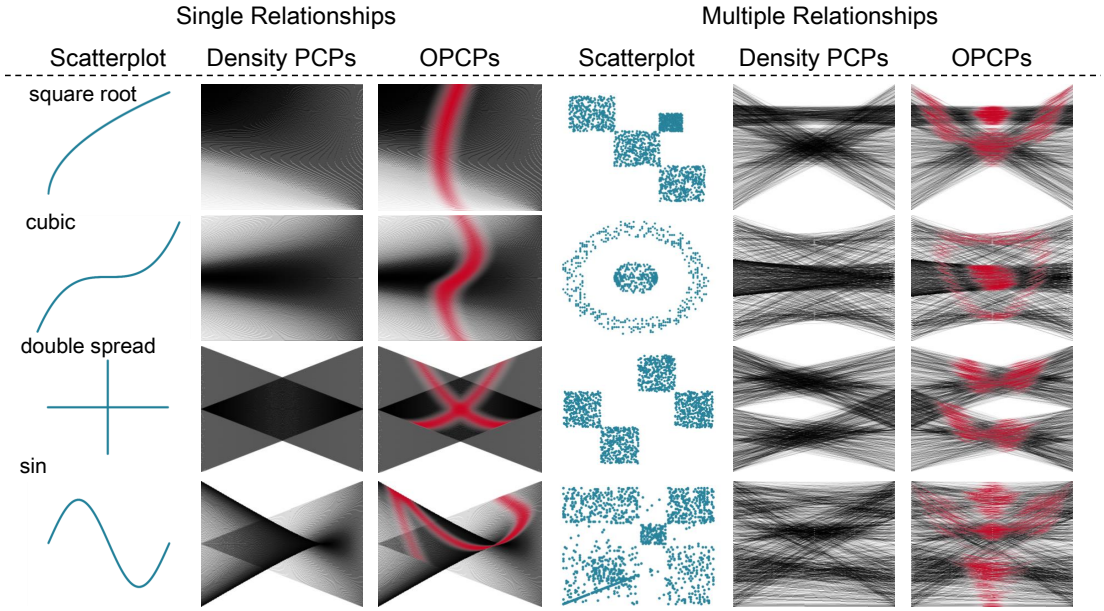


Fig. 14. Examples showing that the OPCPs allow the discernibility of (multiple) patterns or clusters in the synthetic data.

larly to an area brush, applied only in the OPCP space. The O-Prober is useful in cases where the user would like to scan the whole dataset for multiple similar patterns, i.e., lines with a given slope or a range of slopes. The proposed O-Prober requires maximally three interactions, i.e., 2D resizing and translation. Compared to the traditional brushing methods, the O-Brush and O-Prober act only in the OPCP space, hereby allowing for a more precise and local selection and resulting in a reduced amount of required user interaction. The O-Prober and the O-Brush can produce the same result, but their main difference is that the former can be used to probe through the dataset with minimal user interaction, by simply dragging the rectangle over the regions of interest. In our current implementation, the O-Prober is a simple movable rectangle of user-defined size, but it could be easily extended to any arbitrary shape or even a scribbling interface.

Fig. 13 shows a comparison of the O-Brushing methods with alternative PCP brushing methods. It depicts for each brushing technique the best achieved result and the user interactions required to select one specific pattern of interest. In this example, the specific selection is only possible with the composite slider brush and the two proposed O-Brushing methods. However, the O-Brushing methods require fewer user interactions than the composite sliders. In our interactive tool, we included also the state-of-the-art brushes, to enable users to perform selections both in the traditional PCP space and the OPCP space.

4 Results

In this section, we present the results obtained by the application of OPCPs and the O-Brushing to different datasets, intending to provide a deeper understanding into the OPCPs space and its characteristics. To this end, the visualization of the OPCPs was implemented in Python on the GPU, using OpenCL. The interaction for the brushing was realized using the Visualization ToolKit (VTK).

We tested the OPCPs on two different types of data. First, inspired by previous work [22, 43], we tested the behavior of OPCPs on a number of synthetic cases with two-dimensional data with 1000 data points, containing predefined patterns and structures. PCPs and OPCPs are meant for multidimensional data, but we employ these two-dimensional examples for illustration purposes. Secondly, to demonstrate a real usage scenario of OPCPs, we used multivariate data, obtained from various sources [1, 2, 23, 25].

4.1 Results with two-dimensional synthetic stimuli

In Fig. 14, we show our approach as applied to the synthetic stimuli, together with their corresponding scatterplots and Density PCPs. The transformation, i.e., the warping of the scatterplot space on the PCPs that was described in Section 3.2, becomes apparent. We confirmed that the OPCPs facilitate the discernibility of (multiple) data patterns, data outliers and also data structures obstructed by noise compared to

PCPs, which we illustrate with examples, in the following paragraphs.

Discernibility of (multiple) data patterns. We include two main subcategories of relationships in the data: either there is a single relationship in the data, but it is not immediately recognizable, or there are multiple and more complex relationships (Fig. 14). In the first category, we included four different stimuli. For the first two, i.e., cubic and square root, the OPCPs facilitate the identification of the different patterns compared to PCPs, because of the visible correspondance to the scatterplot space. Additionally, the next two, i.e., double spread and sinusoidal, have a similar appearance when shown in the Density PCPs. However, by overlaying the OPCPs, it becomes easier to see that these are different data patterns. Finally, for the second category, it is also easier to identify the multiple relations between the two data dimensions - or data clusters - when employing the OPCPs, as depicted in Fig. 14.

Discernibility of data outliers. We use two synthetic stimuli for which the dimensions are linearly correlated (Fig. 15). However, the second stimulus contains some outliers. By overlaying the OPCPs on the Density PCPs, we can enhance the main pattern in the data, i.e., the linear relationship, without obscuring the outliers.

Discernibility of noise-obstructed data structures. For this case, we created two stimuli with initially no correlation between the two dimensions (Fig. 16). An additional pattern, i.e., a structure with a linear relationship between the dimensions, was then added to the second stimulus. In Density PCPs, this structure is hidden. By overlaying the OPCPs on the Density PCPs, we can visually enhance the obstructed structure in the data and recover the linear relationship.

4.2 Results with multivariate synthetic/real data

In a real-world analysis, PCPs are used to visualize multivariate data. To additionally assess our approach using more complex data with more realistic data patterns across their dimensions, we employ four well-known datasets from various databases. The employed datasets are the `apartments` dataset from [1] with 2290 data points, the `Venus` dataset from [2] with 8784 data points, the `Out5d` dataset from [2] with 16384 data points and the `household` dataset from [25] with 2075259 data points. Fig. 17 shows the results of using OPCPs to represent the above mentioned datasets. The OPCP advantages discussed in 4.1 are again apparent in these cases.

Discernibility of (multiple) data patterns. In the `apartments` dataset, especially between the first two data dimensions, multiple data patterns are emphasized by using OPCPs (Fig. 17-a). This also occurs between the second and third dimension of the same dataset (Fig. 17-b). In the `Venus` dataset, the OPCPs facilitate the identification of distinct patterns, e.g., between the second and third data dimension (Fig. 17-c). Finally, in the `household` dataset, between the first two dimensions, but also between the third and fourth dimensions of the data, patterns that were not visible in the traditional Density PCPs are brought forward with the use of OPCPs (Fig. 17-d,e). Especially, in Fig. 17-e, there are two main patterns in the data, which appear as a single pattern in the Density PCPs.

Discernibility of data outliers. In the `apartments` dataset, OPCPs are able to emphasize outliers, e.g., between the second/third and third/fourth data dimensions, which were not easily discernible in the Density PCPs (Fig. 17-f). In the `Venus` dataset, employing OPCPs allows to visually enhance small patterns in between the first two dimensions (Fig. 17-g), as well as outliers between the third and fourth data dimension that are not visible in the respective Density PCPs (Fig. 17-h). Finally, in the `household` dataset, between the last two data dimensions there are several outliers, which are significantly enhanced with the OPCPs (Fig. 17-i).

Discernibility of noise-obstructed data structures. In the `Out5d` dataset, pattern identification becomes easier throughout all dimensions, especially in parts of the representation, where the patterns are obstructed by noise, e.g., in the last three dimensions of the data (Fig. 17-k,m,n).

Based on the added benefits of OPCPs, it is expected that O-Brushing will facilitate the selection of the respective data structures in the OPCP space, in comparison to state-of-the-art brushing methods, which act in the PCP space, such as the lasso brush, the angular brush and the composite slider brush, as shown in Fig. 13 and 18.

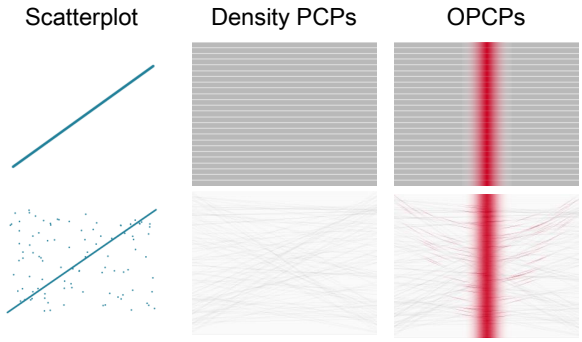


Fig. 15. Examples showing that the OPCPs enable the discernibility of outliers in the synthetic data.

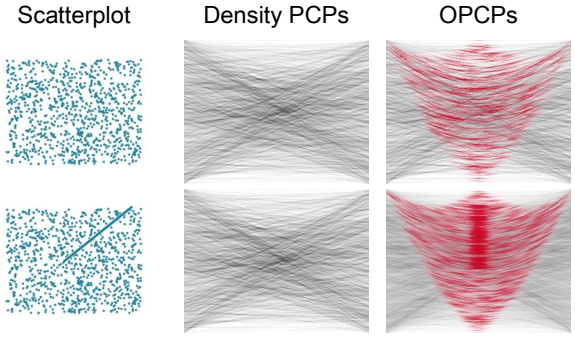


Fig. 16. Examples showing that the OPCPs enable the discernibility of noise-obstructed structures in the synthetic data.

4.3 Performance

The performance of our approach was tested on several datasets from various databases [1, 2, 23, 25]. The datasets vary between 1000 and 2 million data points. The test was conducted on an Alienware Aurora R4 with an Intel Core i7-4820K @ CPU 3.70GHz Processor, 16GB RAM and NVIDIA GeForce GTX 780. The performance results are depicted in Fig. 19. The system is implemented on the GPU and enables interactive brushing. The O-Brush and O-Prober can be employed for almost real-time data-driven selection.

5 Evaluation

To test our approach with respect to the state-of-the-art, we conducted a user evaluation. We used the implemented interactive prototype, which enables visualizing data with Density PCPs and OPCPs, as well as data selection with all five brushing techniques: composite slider brushes, classical lasso brush [17], angular brushing [16], as well as our O-Brush and O-Prober.

The evaluation was designed based on the paper of Lam et al. [24] and consisted of two main parts. The first part was a controlled user study to measure User Performance [24] with the OPCPs and O-Brushing, against Density PCPs and traditional brushing. For this part, we performed three experiments, which are described in detail in the following subsections. The second part consisted of answering a questionnaire to measure User Experience [24], using Likert scales, ranking, and open questions.

We employed 16 participants, with various backgrounds: Computer Science (11, out of which 5 from Computer Graphics and 4 from Visualization), Electrical Engineering (3), Physics (1) and Biomedical Engineering (1). Most of them (9) had preliminary knowledge of PCPs, although only one participant had worked with PCPs before. Before the evaluation, we gave a short introduction, we demonstrated the functionality of the prototype, e.g., how to perform data selections

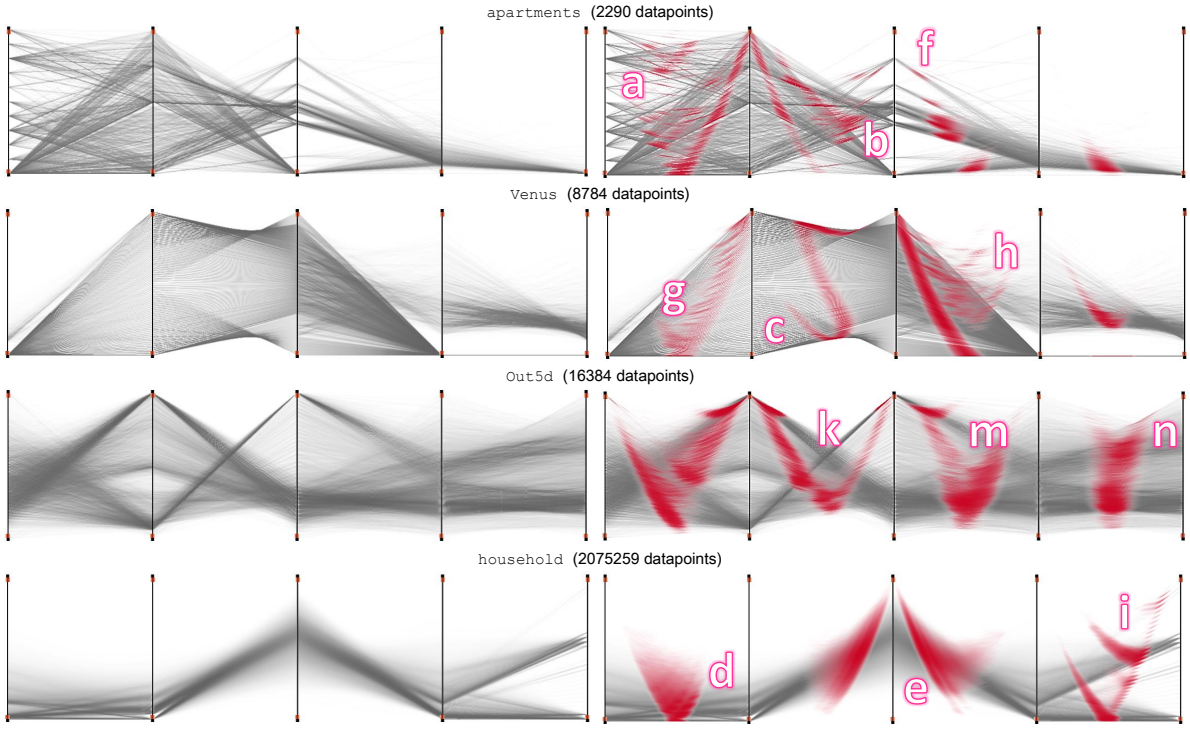


Fig. 17. Application of OPCPs to multivariate synthetic or real data obtained from various databases [1, 2, 25].

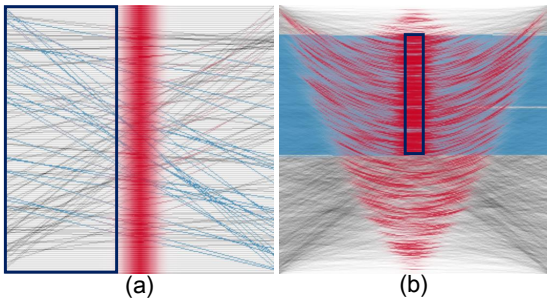


Fig. 18. Examples showing that the OPCPs allow the selection of (a) outliers and (b) noise-obstructed structures in the data.

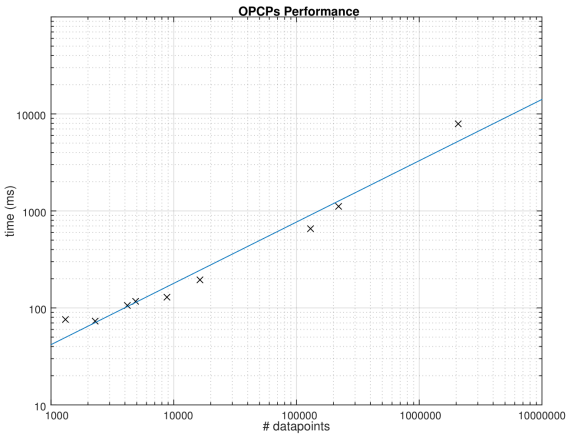


Fig. 19. Performance times of OPCPs for multivariate synthetic or real data from various databases [1, 2, 23, 25].

with each method, and we allowed participants to use it, until they felt confident with it. In average, people spent around 5 mins on the prototype before the experiment.

5.1 First Part: User Performance

We performed three experiments. The *first experiment* aimed at measuring the performance of users in discerning (1) patterns, (2) outliers, and (3) data structures obstructed by noise using Density PCPs or OPCPs. We created two comparable, two-dimensional synthetic datasets per case and we visualized them with both representations. Then, we showed static images of the representations to the users in a randomized order, and we asked them to perform tasks such as identifying and pointing out patterns in the data, outliers and noise-obstructed data structures. For each one of the static images, we measured the time that users needed to give a conclusive answer and accuracy of their answers. Since the data were synthetic, we already knew the exact number of, e.g., patterns in the data. Thus, every wrong or unidentified pattern was penalized in the accuracy measurement.

The *second experiment* aimed at measuring user performance in selecting (1) specific patterns, (2) outliers, and (3) noise-obstructed data structures, with state-of-the-art brushing or O-Brushing. We created a two-dimensional synthetic dataset per case. We showed static images to the users, explaining which part of the data needed to be selected. Then, we asked them to perform a selection of the previously specified data part, using only one of the brushing techniques at a time, in a random order. All tasks were possible with all methods. Again, we measured time, accuracy, and number of interactions, i.e., number of clicks, required for task completion.

The *third experiment* aimed at measuring performance with multivariate, complex data and tasks. We created two comparable five-dimensional synthetic datasets, using the PCDC tool [7]. Then, we designed a set of questions, which were related to identifying and/or selecting data patterns, outliers and noise-obstructed structures. The users were asked to apply traditional brushing to one of the datasets with PCPs, and O-Brushing to the other dataset with OPCPs to perform the given tasks. The order of the dataset and approach, as well as their combination, was alternated randomly to reduce bias from learning. We measured completion time, accuracy and number of interactions, i.e., clicks required from the user, for the task completion.

The outcome of the statistical analysis of the experiments is summarized in Fig. 20. The first and third experiment were analyzed with Paired t-tests, while the second was analyzed with ANOVA and

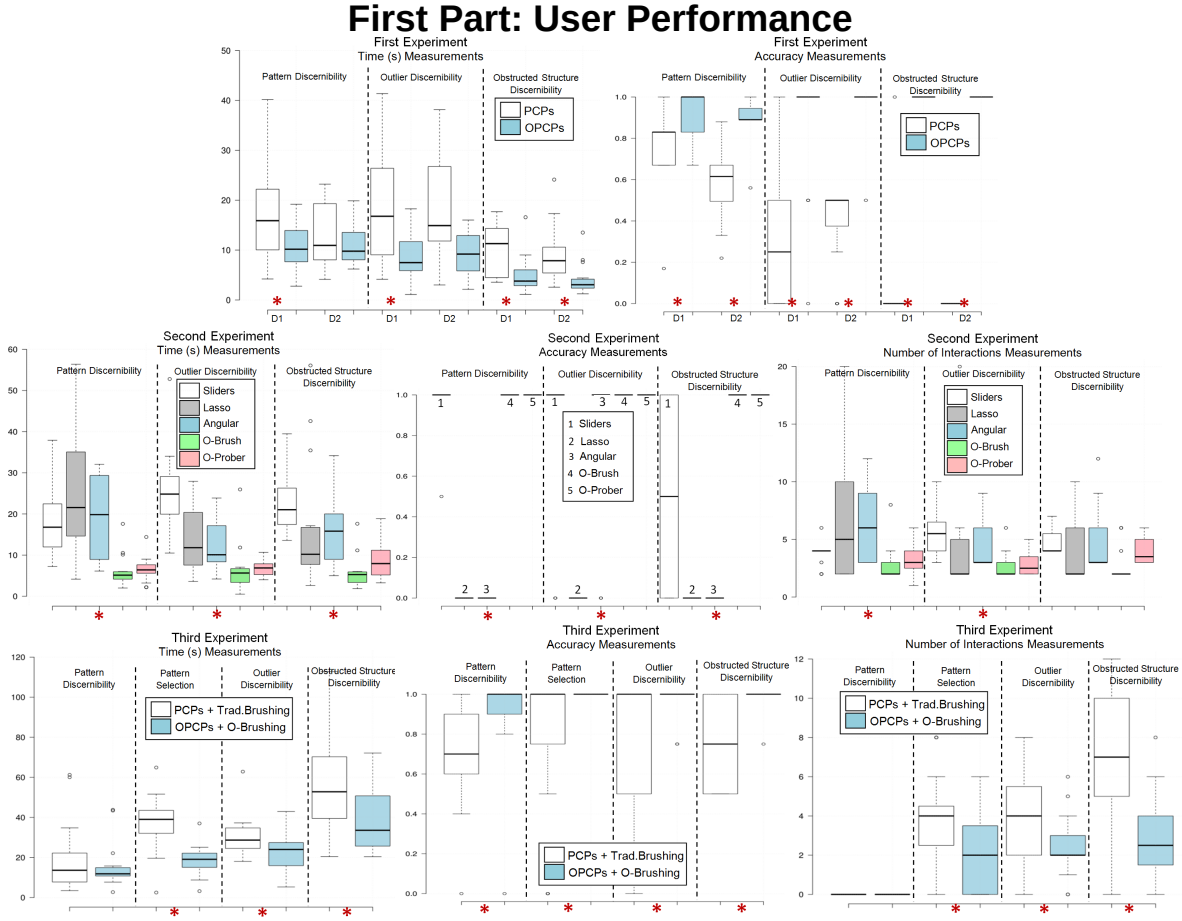


Fig. 20. Results for the experiments conducted as part of the evaluation, for the User Performance part. The small white circles denote outliers in the measurements. The asterisks denote a statistically significant difference (ρ <0.05) between the measurements, as it resulted from our statistical analysis using ANOVA and Tukey's HSD test.

Tukey's HSD test, for statistical significance. The results of the *first experiment* indicate that identification of patterns, outliers and noise-obstructed structures is more accurate with OPCPs than with PCPs (ρ<0.01). Especially, in case where a structure is obstructed by noise in the data, the OPCPs were much more accurate ($\mu = 1, \sigma = 0$) than PCPs ($\mu = 0.06, \sigma = 0.25$). The distinction of noise-obstructed structures is also faster (ρ<0.05) in OPCPs: users required half the time to recognize this kind of structures in the data with OPCPs than with PCPs. For pattern and outlier detection, there is no conclusive result for the time performance, but the accuracy is significantly improved with OPCPs. The outcome of the *second experiment* shows that O-Brushing is faster and more accurate (ρ<0.01) in all cases. For pattern and outlier selection, O-Brushing also requires significantly less interactions (ρ<0.05). From Tukey's HSD test, it results that there is no statistically significant difference between the performance of users when using the O-Brush or the O-Prober. Based on this test, the overall ranking of the different brushing techniques for the three investigated tasks results as: the two variants of O-Brushing, angular brushing, composite brushing using sliders and lasso brushing. The results of the *third experiment* demonstrate that our approach is more accurate than the state-of-the-art approach for the four given tasks. The combined use of OPCs with O-Brushing had an average accuracy of 0.96 for all tasks, while traditional PCPs with standard brushing only 0.75. In this experiment, there were no indications that pattern discernibility requires less time with OPCPs. However, for the other three tasks the use of OPCPs and the proposed O-Brushing makes a big difference in performance times. For example, for pattern selection our approach requires half the time of the traditional approach. Overall, there is an indication that when selection is involved, our approach is also significantly faster and requires less interaction (ρ<0.05).

5.2 Second Part: User Experience

The second part of the evaluation consisted of conducting a survey. First, we asked users to grade PCPs and OPCPs and, also, the five previously used brushing methods, using *Likert scales*. The outcome of the statistical analysis of the experiments is summarized in Fig. 21. From the statistical analysis, it resulted that the PCPs were easier to understand, but the OPCPs were considered significantly easier to use (4.13), more useful (4.44) and also more suitable for the identification of patterns (4.44), outliers (4.38) and noise-obstructed data structures (4.31), compared to traditional PCPs. Moreover, the composite sliders and O-Brushing were considered easier to use and useful, while the easiest to understand were composite sliders and the O-Prober. The sliders and O-Brushing were considered most suitable for pattern selection, while for outlier and obstructed structure selection only O-Brushing was preferred. The next part of the questionnaire consisted on ranking the two representations using the same scale and the five brushing methods. The OPCPs were ranked significantly higher (8.31) than the PCPs (5.81) (ρ<0.05), while the O-Brush and the O-Prober were ranked significantly higher (8.25 and 8.29, respectively) than the sliders (6.50), the lasso (4.81) and the angular brushing (5.25) (ρ<<0.01). The questionnaire was concluded with open questions. The evaluation participants replied that the OPCPs "*can be very strong in structure detection in the data*", especially "*when there is a lot of overlap in the data*". However, "*the OPCPs take more time to get used to*" and "*might require some training for naïve users*". Also, finding "*simple correlations across dimensions can be easier sometimes with PCPs only*". O-Brushing makes it "*easier to select patterns locally*", but "*O-Prober could be improved by using also different shapes, other than the rectangle*". Most users commented that our approach supported them more in the identification and selection of patterns and

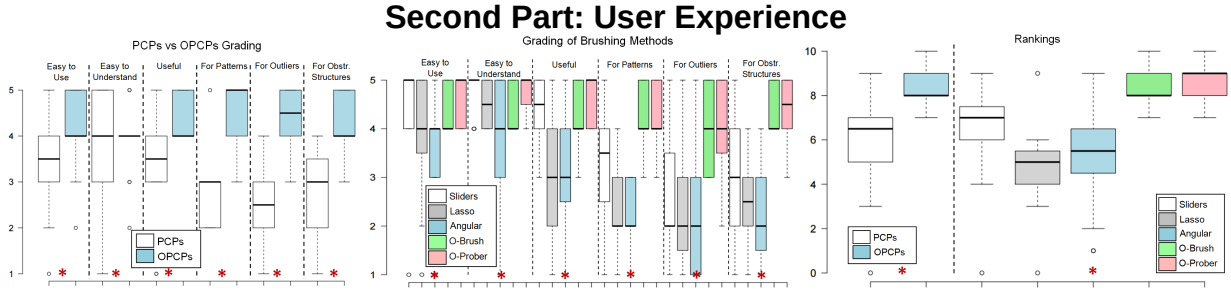


Fig. 21. Results for the experiments conducted as part of the evaluation, for the User Experience part. The asterisks denote a statistically significant difference ($\rho < 0.05$) between the measurements, as it resulted from our statistical analysis using ANOVA and Tukey's HSD test.

outliers, in particular. For simple cases, due to the fact that OPCPs require prior familiarization and training, they might be less suitable. However, for cluttered data, the advantages are straightforward.

6 Discussion

The results of the application of OPCPs on synthetic and real datasets presented in Sec. 4, as well as the evaluation results of Sec. 5 raised several points for discussion and limitations.

Firstly, the proposed OPCPs are a visual enhancement of PCPs for more efficient discernibility of patterns, outliers and noise-obstructed structures in the data. In the paper of Holten et al. [18], it is stated that combining scatterplots with PCPs can result in significant performance gains for the users. In many papers, the combination of PCPs with scatterplots is limited to having multiple linked views, where interactive linking and brushing can reflect selections from one representation to the other. However, in this case, users need to switch between windows and use their mental memory for data exploration and analysis, e.g., when the user performs an operation and sees the result in another window. Our OPCPs, instead, are not aiming at substituting scatterplots or at using linked scatterplot views. They focus on giving a better understanding of the data represented by PCPs, by integrating in a seamless way the two representations in one, combining their benefits and reducing the memory limitations that result from switching between the two separate representations. We consider that a comparison between scatterplots linked to PCPs against solely OPCPs is out of the scope of this paper, as the latter is a visual enhancement of PCPs and not a new representation on its own. In fact, as noticed by Holten et al. [18], PCPs might still be better than scatterplots in showing the actual shape of clusters, which is evident in our visualization. Still, OPCPs could be combined with linked scatterplots and it would also be interesting to investigate a comparison between scatterplots linked to PCPs and scatterplots linked to OPCPs.

Additionally, from the evaluation, it resulted that the interpretability of the patterns might not be straightforward and requires a certain level of familiarization with the enhancement. However, during the evaluation, the users were able to identify patterns more accurately than PCPs. Also, the cognitive load of OPCPs is not so significant to slow down the analysis of the data. As it can be seen in the evaluation results, in the vast majority of the tasks, the time needed to perform an operation using OPCPs and the related O-Brushing is significantly less than the time needed to perform the same operation with the state-of-the-art techniques. This is a first indication that the interpretability of patterns in OPCPs is not compromised. In a future additional evaluation, it would be interesting to research this further.

Moreover, so far, there was no evidence in the user evaluation that the use of OPCPs might be distractive for the user or interfering with bundle tracking, which is the main advantage of the use of PCPs. OPCPs are indeed a new visual enhancement that requires some training, as pointed out by users. However, in our interactive tool the appearance of OPCPs can be adjusted by fine-tuning the σ and γ values to make the enhancement as prominent as the user would like. Also, there is always the option to adjust the color and opacity of the OPCP segments, to interfere less with the underlying PCPs and the polyline bundles. In the user evaluation, we included tasks where bundle tracking was necessary. In these cases, the users could perform the tasks without problems. However, for a more conclusive answer to this point, a more extensive study would be needed.

For the brushing functionality, in the interactive version of our tool, the user can select in which of the two spaces, i.e., PCP space or OPCP space, he/she would like to brush. In the OPCP space, the two proposed O-Brushing methods can be employed, while in the PCP space, state-of-the-art brushing, such as angular or lasso or composite brushing, can be used. As some users stated during the evaluation, having this possibility to choose the space to perform selections on the data is useful in different occasions: for example, if the user needs to perform selections based on the range values of some dimensions, the state-of-art brushing methods are more appropriate and more straightforward to use. However, if specific patterns, or outliers or structures in the data need to be selected, then O-Brushing is more efficient.

Limitations. We foresee some limitations of our approach. First, OPCPs require some familiarization, as they are not immediately intuitive. Additionally, they require a wider spacing between the dimension axes as compared to traditional PCPs in order to be effective. Moreover, the OPCPs should be accompanied by PCPs, to preserve context and connectivity across dimensions. Finally, the O-Prober could improve by using free-hand shapes or a scribbling interface instead of the predefined rectangle. This would enable easier, faster and more accurate selection of specific patterns with OPCPs, similarly to the shape-based method by Muigg et al. [28] for traditional brushing.

7 Conclusions and Future Work

Parallel Coordinate Plots exhibit overplotting, which results in a cluttered view on the data. Therefore, discerning the underlying data information and selecting interesting patterns can become difficult. We proposed a new technique, the Orientation-enhanced Parallel Coordinate Plots, to improve the view and discernibility of patterns in otherwise cluttered PCPs. We achieved our goal by visually enhancing parts of each PCP line with respect to its slope, hereby incorporating information from 2D scatterplots in the representation [18]. Compared to the state-of-the-art, our approach is simple and provides better discernibility of data patterns, especially when there are multiple overlapping patterns or when there are outliers and structures, obstructed by noise. We evaluated our approach with several synthetic and real-world datasets. One of the main advantages of OPCs is that thy allow a new and versatile selection method, the Orientation-enhanced Brushing. Brushing in the OPCPs space enables an efficient selection of individual data structures involving a reduced user interaction when compared to the state-of-the-art selection tools in PCPs. On the other hand, OPCPs require more training, compared to PCPs.

A direction for future work includes employing color transfer functions in the OPCPs for better discrimination of the different data patterns, or even clustering. Moreover, it would be interesting to extend the evaluation of our proposed visual enhancement, but also of the related brushing method, to cover the points discussed in Sec. 6. Finally, the extension of the O-Prober to other shapes should allow easier, faster, and more interactive selections of data patterns.

8 Acknowledgements

This work was partially supported by the FP7 European Project Harvest4D and the FP7 European Project DR THERAPAT.

REFERENCES

[1] Institute for Information Systems at the TU Braunschweig (Germany).

[2] XmdvTool Datasets: http://davis.wpi.edu/xmdv/datasets.html.

[3] D. F. Andrews. Plots of high-dimensional data. *Biometrics*, pages 125–136, 1972.

[4] M. Ankerst, S. Berchtold, and D. A. Keim. Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In *Information Visualization, 1998. Proceedings. IEEE Symposium on*, pages 52–60. IEEE, 1998.

[5] A. O. Artero. Uncovering clusters in crowded parallel coordinates visualizations. In *In Proceedings of the IEEE Symposium on Information Visualization 2004*, pages 81–88. IEEE Computer Society, 2004.

[6] E. Bertini, L. Dell'Aquila, and G. Santucci. Springview: Cooperation of radviz and parallel coordinates for view optimization and clutter reduction. In *Coordinated and Multiple Views in Exploratory Visualization, 2005.(CMV 2005). Proceedings. Third International Conference on*, pages 22–29. IEEE, 2005.

[7] S. Bremm, T. Von Landesberger, M. Heß, and D. Fellner. PCDC-On the highway to data-a tool for the fast generation of large synthetic data sets. In *EuroVis Workshop on Visual Analytics*, pages 7–11, 2012.

[8] G. Ellis and A. Dix. Enabling automatic clutter reduction in parallel coordinate plots. *Visualization and Computer Graphics, IEEE Transactions on*, 12(5):717–724, 2006.

[9] G. Ellis and A. Dix. A taxonomy of clutter reduction for information visualisation. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1216–1223, 2007.

[10] E. Fanea, S. Carpendale, and T. Isenberg. An interactive 3D integration of parallel coordinates and star glyphs. In *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*, pages 149–156. IEEE, 2005.

[11] Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *Proceedings of the conference on Visualization'99: celebrating ten years*, pages 43–50. IEEE Computer Society Press, 1999.

[12] Z. Geng, Z. Peng, R. S. Laramee, J. C. Roberts, and R. Walker. Angular histograms: Frequency-based visualizations for large, high dimensional data. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2572–2580, 2011.

[13] R. C. Gonzalez and R. E. Woods. *Digital Image Processing (2nd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2002.

[14] M. Graham and J. Kennedy. Using curves to enhance parallel coordinate visualisations. In *Information Visualization, 2003. IV 2003. Proceedings. Seventh International Conference on*, pages 10–16. IEEE, 2003.

[15] L. Harrison, F. Yang, S. Franconeri, and R. Chang. Ranking visualizations of correlation using weber's law. *Visualization and Computer Graphics, IEEE Transactions on*, 20(12):1943–1952, 2014.

[16] H. Hauser, F. Ledermann, and H. Doleisch. Angular brushing of extended parallel coordinates. In *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*, pages 127–130. IEEE, 2002.

[17] J. Heinrich and D. Weiskopf. State of the Art of Parallel Coordinates. 2012.

[18] D. Holten and J. J. Van Wijk. Evaluation of cluster identification performance for different pcp variants. In *Computer Graphics Forum*, volume 29, pages 793–802. Wiley Online Library, 2010.

[19] C. Hurley and R. Oldford. Pairwise display of high-dimensional information via eulerian tours and hamiltonian decompositions. *Journal of Computational and Graphical Statistics*, 19(4), 2010.

[20] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(2):69–91, 1985.

[21] A. Inselberg. *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. Springer-Verlag New York, 2009.

[22] J. Johansson, P. Ljung, M. Jern, and M. Cooper. Revealing structure within clustered parallel coordinates displays. In *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*, pages 125–132. IEEE, 2005.

[23] S. Johansson and J. Johansson. Interactive dimensionality reduction through user-defined combinations of quality metrics. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):993–1000, 2009.

[24] H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale. Empirical studies in information visualization: Seven scenarios. *Visualization and Computer Graphics, IEEE Transactions on*, 18(9):1520–1536, 2012.

[25] M. Lichman. UCI machine learning repository, 2013.

[26] A. R. Martin and M. O. Ward. High dimensional brushing for interactive exploration of multivariate data. In *Proceedings of the 6th Conference on Visualization'95*, page 271. IEEE Computer Society, 1995.

[27] K. T. McDonnell and K. Mueller. Illustrative parallel coordinates. In *Computer Graphics Forum*, volume 27, pages 1031–1038. Wiley Online Library, 2008.

[28] P. Muigg, J. Kehrer, S. Oeltze, H. Piringer, H. Doleisch, B. Preim, and H. Hauser. A Four-level Focus+ Context Approach to Interactive Visual Analysis of Temporal Features in Large Scientific Data. In *Computer Graphics Forum*, volume 27, pages 775–782. Wiley Online Library, 2008.

[29] M. Novotny and H. Hauser. Outlier-preserving focus+ context visualization in parallel coordinates. *Visualization and Computer Graphics, IEEE Transactions on*, 12(5):893–900, 2006.

[30] W. Peng, M. O. Ward, and E. A. Rundensteiner. Clutter reduction in multi-dimensional data visualization using dimension reordering. In *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*, pages 89–96. IEEE, 2004.

[31] T. Porter and T. Duff. Compositing digital images. In *ACM Siggraph Computer Graphics*, volume 18, pages 253–259. ACM, 1984.

[32] P. L. Rheingans. Task-based color scale design. In *28th AIPR Workshop: 3D Visualization for Data Exploration and Decision Making*, pages 35–43. International Society for Optics and Photonics, 2000.

[33] H. Siirtola. Direct manipulation of parallel coordinates. In *Information Visualization, 2000. Proceedings. IEEE International Conference on*, pages 373–378. IEEE, 2000.

[34] H. Siirtola and K.-J. Räihä. Interacting with parallel coordinates. *Interacting with Computers*, 18(6):1278–1309, 2006.

[35] H. Theisel. Higher Order Parallel Coordinates. In *VMV*, pages 415–420, 2000.

[36] A. Unwin, C. Volinsky, and S. Winkler. Parallel coordinates for exploratory modelling analysis. *Computational Statistics & Data Analysis*, 43(4):553–564, 2003.

[37] M. O. Ward. XmdvTool: Integrating Multiple Methods for Visualizing Multivariate Data. In *Proceedings of the Conference on Visualization '94*, VIS '94, pages 326–333, Los Alamitos, CA, USA, 1994. IEEE Computer Society Press.

[38] M. O. Ward. Creating and manipulating n-dimensional brushes. In *Proceedings of Joint Statistical Meeting*, pages 6–14. Citeseer, 1997.

[39] E. J. Wegman. Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association*, 85(411):664–675, 1990.

[40] N. Wong, S. Carpendale, and S. Greenberg. Edgelens: An interactive method for managing edge congestion in graphs. In *Information Visualization, 2003. INFOVIS 2003. IEEE Symposium on*, pages 51–58. IEEE, 2003.

[41] P. C. Wong and R. D. Bergeron. Multiresolution multidimensional wavelet brushing. In *Visualization'96. Proceedings.*, pages 141–148. IEEE, 1996.

[42] J. Yang, W. Peng, M. O. Ward, and E. A. Rundensteiner. Interactive hierarchical dimension ordering, spacing and filtering for exploration of high dimensional datasets. In *Information Visualization, 2003. INFOVIS 2003. IEEE Symposium on*, pages 105–112. IEEE, 2003.

[43] X. Yuan, P. Guo, H. Xiao, H. Zhou, and H. Qu. Scattering points in parallel coordinates. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):1001–1008, 2009.

[44] H. Zhou, W. Cui, H. Qu, Y. Wu, X. Yuan, and W. Zhuo. Splatting the lines in parallel coordinates. In *Computer Graphics Forum*, volume 28, pages 759–766. Wiley Online Library, 2009.

[45] H. Zhou, X. Yuan, H. Qu, W. Cui, and B. Chen. Visual clustering in parallel coordinates. In *Computer Graphics Forum*, volume 27, pages 1047–1054. Wiley Online Library, 2008.

Mitig Adapt Strateg Glob Change
DOI 10.1007/s11027-015-9651-2

ORIGINAL ARTICLE

An interactive simulation and visualization tool for flood analysis usable for practitioners

Johannes G. Leskens[1] · Christian Kehl[2] · Tim Tutenel[2] · Timothy Kol[2] · Gerwin de Haan[2] · Guus Stelling[3] · Elmar Eisemann[2]

Received: 11 May 2014 / Accepted: 25 March 2015
© The Author(s) 2015. This article is published with open access at Springerlink.com

Abstract Developing strategies to mitigate or to adapt to the threats of floods is an important topic in the context of climate changes. Many of the world's cities are endangered due to rising ocean levels and changing precipitation patterns. It is therefore crucial to develop analytical tools that allow us to evaluate the threats of floods and to investigate the influence of mitigation and adaptation measures, such as stronger dikes, adaptive spatial planning, and flood disaster plans. Up until the present, analytical tools have only been accessible to

✉ Johannes G. Leskens
j.g.leskens@utwente.nl

Christian Kehl
christian.kehl@uni.no

Tim Tutenel
tim.tutenel@gmail.com

Timothy Kol
trkol@tudelft.nl

Gerwin de Haan
info@gerwindehaan.nl

Guus Stelling
stellinghydraulics@hotmail.com

Elmar Eisemann
e.eisemann@tudelf.nl

[1] Water Engineering and Management, University of Twente, PO Box 217, 7500 AE Enschede, The Netherlands

[2] Computer Graphics & Visualization, Delft University of Technology, Mekelweg 4, 2628CD Delft, The Netherlands

[3] Computational Hydraulics, Stelling Hydraulics, Venezuelastraat 12, 2622 BR Delft, The Netherlands

Published online: 29 May 2015

domain experts, as the involved simulation processes are complex and rely on computational and data-intensive models. Outputs of these analytical tools are presented to practitioners (i.e., policy analysts and political decision-makers) on maps or in graphical user interfaces. In practice, this output is only used in limited measure because practitioners often have different information requirements or do not trust the direct outcome. Nonetheless, literature indicates that a closer collaboration between domain experts and practitioners can ensure that the information requirements of practitioners are better aligned with the opportunities and limitations of analytical tools. The objective of our work is to present a step forward in the effort to make analytical tools in flood management accessible for practitioners to support this collaboration between domain experts and practitioners. Our system allows the user to interactively control the simulation process (addition of water sources or influence of rainfall), while a realistic visualization allows the user to mentally map the results onto the real world. We have developed several novel algorithms to present and interact with flood data. We explain the technologies, discuss their necessity alongside test cases, and introduce a user study to analyze the reactions of practitioners to our system. We conclude that, despite the complexity of flood simulation models and the size of the involved data sets, our system is accessible for practitioners of flood management so that they can carry out flood simulations together with domain experts in interactive work sessions. Therefore, this work has the potential to significantly change the decision-making process and may become an important asset in choosing sustainable flood mitigations and adaptation strategies.

Keywords Flood · Visualization · Decision-making · Large-scale rendering

1 Introduction

Climate changes already have drastic implications for rainfall and ocean levels, and the situation is likely to worsen (Barros et al. 2014). Over 50 % of the world population lives in cities (WHO 2013) and more than two thirds of the largest cities are vulnerable to rising sea levels as a result of climate change (McGranahan et al. 2007). Hence, millions of people are therefore exposed to the risk of extreme floods and storms.

Developing strategies to mitigate or to adapt to the threats of floods is an important topic in the context of climate changes. Several countries and unions, including the European Union (EU), have put a multi-layer safety approach into place as a framework for the development of these mitigation and adaptation strategies (EU 2005). The multi-level safety approach consists of three layers. Layer 1 focuses on protection measures in the form of levees and dikes and has traditionally received most attention and funding (Kabat et al. 2009). For example, in the Netherlands, a 5-year maintenance cycle as well as a standardization for reinforcements has been established (STOWA 2008). Layer 2 consists of waterproof spatial planning, while layer 3 considers disaster management. Examples of mitigation measures through spatial planning (layer 2) are local protection of hospitals, schools, and utility companies or spatial planning measures that improve the chances of evacuation from a flooded area. Flood disaster management (layer 3) can be improved by setting up disaster protocols and practicing decision-making under worst-case scenarios.

One of the main challenges for following a multi-level safety approach is the integrated decision-making process it requires. This means that different practitioners, related to the three layers mentioned above, have to collaborate and find integrated solutions for

flood management. These practitioners include policy analysts or political decision-makers from municipalities, water boards, and provinces or representatives from fire and police departments, hospitals, and energy companies.

In this process, it is important to build the capacity among these practitioners for involvement in flood management. Analytical tools, e.g., Sobek (http://www.deltaressystems.com/hydro/product/108282/sobek-suite) or Mike11 (http://www.mikebydhi.com), can be used to gain insight into the consequences of floods under different climate scenarios or alternative measures in terms of water depths, flow velocities, or damages. They offer predictions by simulating the physical processes involving various area features such as elevation and roughness resistance and external forces such as storm events and dam breaches (Al-Sabhan et al. 2003; Bates and Roo 2000; Moel and Aerts 2011; Stelling 2012a). To address model uncertainties, analytical tools are normally validated with historical events or when such measurements are not available, a sensitivity analysis or ensemble calculations can be carried out (Walker et al. 2003).

Up to the present, analytical tools have only been accessible to domain experts, such as hydraulic engineers and modelers, as the involved simulation processes are complex and rely on computational and data-intensive models. Practitioners (i.e., policy analysts and political decision-makers) are usually only presented with resulting maps. Typically, these indicate the impacts of individual flood hazards or the results of risk analyses, in which multiple hazard scenario's, each having a certain likelihood, are combined into risk maps (Apel et al. 2004).

Despite the advantages that these maps may provide for practitioners of flood management, poor use is often made of this information for supporting decisions (Leskens et al. 2014b). Mors et al. (2005) show that practitioners of flood management, operating under regulatory, institutional, political, resource, and other constraints, prioritize other concerns over more sophisticated maps with model information about flood risks. An underlying reason might be the difference in perception of the threats of floods between domain experts, such as modelers or hydraulic engineers, and practitioners (Faulkner et al. 2007; Janssen et al. 2010; Timmerman et al. 2010; Wood et al. 2012). Domain experts generally frame flood issues using scientific knowledge and expertise. They assume that with more detailed model, information analyses will improve and better decisions can be made. Practitioners, on the other hand, often lack the capacity and time to incorporate the results of these complex analyses in their decisions.

A way to better support practitioners with the output of analytical tools is to improve their involvement in the application of these tools (Voinov and Bousquet 2010; Leskens et al. 2014b). This involvement can improve the alignment of the information requirements of practitioners with the opportunities and limitations of analytical tools (Leskens et al. 2014a). However, as mentioned, the existing analytical tools are only accessible to domain experts, as the involved simulation processes are complex and rely on computational and data-intensive models. These analytical tools have specialized interfaces and their application—for example, to picture a future flood scenario—takes multiple hours (Leskens et al. 2014b).

In this paper, we present a system that is focused on making the use of flood analysis tools accessible for practitioners of flood management, such that they can carry out flood analysis together with domain experts in interactive work sessions. The system gives the user the possibility to test several disaster scenarios and to receive direct visual feedback. It provides realistic images to help practitioners in interpreting these outputs of the system. We conducted a user study to test the effectiveness of this interface

and its outputs. In order to investigate the applicability in practice, we illustrated the use of our system for real-world data in a case study for the area of West-Friesland, The Netherlands.

This work makes a two-fold contribution: 1) we present a working system and show the utility of our approach, and 2) we explain the technical contributions needed to achieve a working solution. The paper offers an overview of our system and its possibilities, focusing on interaction and visualization. We also give a description of the novel techniques that have been used. The results of two tests are presented.

2 Our system

In this section, we explain how the system was developed and give an overview of the key features and their technical backgrounds. For more detailed descriptions, we provide references to other publications.

2.1 Development of the system

The system presented here builds upon a prototype developed between 2011 and 2014, involving researchers and engineers from Delft University of Technology, Deltares, and Nelen & Schuurmans.

The key features of the system are based on user requirements that were derived from 13 semi-structured interviews among policy analysts of the regional water board Hoogheemraadschap van Delfland (The Netherlands), conducted in 2011 and reported in Leskens et al. (2014a). The interviews focused specifically on three questions. (1) What is your task or role in decision-making processes? (2) What information do you require to carry out this task? (3) What functions do you need from an analysis tool that can be operated during a work session? From these 13 interviews, it emerged that the following capabilities were considered important for using an analysis tool during a work session with domain experts: (1) technical reliability, (2) the possibility to assess the effectiveness of multiple scenarios within the time horizon of a work session, and (3) understandable output for non-water specialists.

Based on these user requirements, the key features of our system focus on a realistic visualization of floods and interactivity to enable practitioners to explore various options for flood mitigation and adaptation measures rapidly, together with domain experts. To ensure technical reliability (i.e., the first user requirement), our system combines a high spatial resolution (i.e., 0.5 m by 0.5 m) and the inclusion of all relevant processes (i.e., overland flow, groundwater flow, canal flow, and sewer flow) (Stelling 2012b).

There are other flood analysis tools available that provide comparable features and user interfaces (Nóbrega et al. 2008; Bates and De Roo 2000). However, our system is unique in providing the distinct features in the required scale (>1000 km^2) and resolution (>10 height values per square meter). Making an interactive system with a realistic visualization for such large areas is a very challenging goal.

2.2 Overview

Three dimensional (3D) visualization Our 3D visualization is aimed at supporting practitioners in interpreting and understanding the impact of simulated flood hazards. It shows the properties of simulation results in three spatial dimensions and a realistic rendering,

mapped to real-world phenomena. In respect to two dimensional (2D) visualizations, in which the impacts of flood hazards can only be viewed directly from above, the 3D visualization shows the impacts also from the side. Flood depths can therefore be examined without the use of a legend in which typically different shadings of blue are used to indicate the depth of a flood. The basis for this realistic 3D visualization is data from Light detection and range (LiDAR) scans coupled to output of water simulations. LiDAR data is collected by a technology that measures distance by illuminating a target with a laser and analyzing the reflected light. The result of these LiDAR scans, usually carried out by helicopters, are points specified in three spatial dimensions (x, y, z), with a resolution of around 15–50 points per square meter. These points, colored via aerial images, ensure recognition of the area and its features (e.g., houses, cars, and trees) in high detail. Water properties, based on the simulations, are projected realistically into this 3D visualization. For example, flow directions are visualized by moving waves and water depth is displayed by adjusting the light extinction. Furthermore, the system supports stereo rendering to create the illusion of depth, if the corresponding equipment is available. Extensive interactive device support gives users the possibility to navigate the 3D world with ease. This also allows us to adapt the system's navigation to the cognitive and motoric capacities of different audiences (e.g., general public, museum visitors, decision-makers, managers, hydrologists, etc.). Our system builds upon an efficient out-of-core rendering system that displays large-scale light detection and ranging (LiDAR) data (Haan 2009, 2010; Kehl and de Haan 2012). Even though large-scale point-based rendering is a classic rendering topic (Dachsbacher et al. 2003; Rusinkiewicz and Levoy 2000), a solution that combines large-scale water and terrain visualizations was previously unavailable.

Interaction For interaction with the simulation process, our system relies on a simulation display that shows the map of the study area (Fig. 1). Different from the 3D visualization system, as presented in the former section, the study area in the simulation

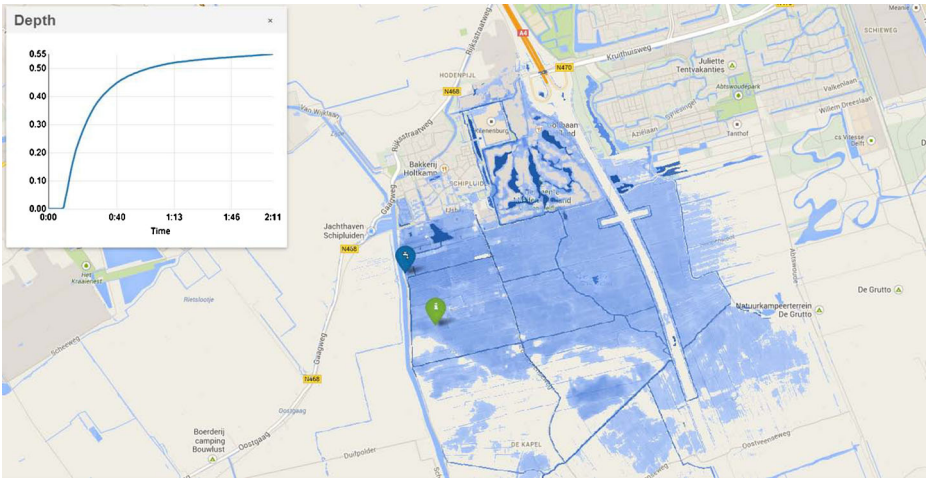


Fig. 1 A screenshot of the 2D interface used for interaction

display can only be viewed directly from above. Once a simulation is carried out, the outcome can be presented in the 3D visualization system and be viewed by a helicopter perspective.

A user can choose two different methods to add water flows to the study area. First, a point source can be added, consisting of a flow discharge or a water level. This point source can, for example, be used to simulate a dam breach or a leaking sewer pipe. Second, a spatially distributed source can be added, consisting of a flow discharge to a selected area, which can be used to simulate rainfall events. Several properties can be selected for both types of water sources, for example, whether the flow discharge of a point source is constant or time-dependent, or whether the distributed discharge has a circular or an irregular shape. The impact of these sources on the area can be assessed simultaneously with the simulation process. The water depths are indicated by a dynamic blue overlay and can also be assessed by a graph on random points or over a random cross section. Flow velocity is indicated by bubbles in the water.

Aside from the various options to add water sources, the properties of the study area's terrain can be adjusted as well. This can be done in three ways. First, the elevation of the area can be locally increased or decreased—for example, to simulate the effectiveness of digging passages in local elevations that obstruct water flowing to storage areas or to test the effectiveness of elevating levees with sand bags. Second, the land use type can be changed—for example, to increase the rainwater infiltration capacity of the ground by changing paved areas into unpaved areas. Third, the water system in the study area, consisting of canals, pumps, weirs, and culverts, can be adjusted—for example, to test the effectiveness of stronger pumps, broader canals, or higher weirs.

Adding water sources to the study area and adjusting the study area itself can be done at any time by the user during the simulation process. The simulation can also be stopped to make adjustments to the water sources or the study area, after which the process is resumed. The display is made accessible via an internet browser. Therefore, users do not need to install special software on their computers. In case multiple users access the system at the same time, one of the users has the authority to make adjustments to the simulation. This depends on the pre-specified user rights.

The interactive character of the simulation display, as described above, is made possible by a strong computation core. This computation core consists of numerical methods to solve the 2D hydraulic equations for water flow, under the conditions of short computation times while still conserving the details of the study area. The technical details of this flood simulation model can be found in Stelling (2012b) and Casulli and Stelling (2013).

2.3 Technical contributions

Several hurdles were overcome to obtain a system that is accessible to practitioners who are no domain experts. We added four extensions to the existing 3D visualization techniques (Haan 2010; Kehl and de Haan 2012; Kehl et al. 2013). First, we introduce a solution to directly manipulate the LiDAR point sets to improve collaborative aspects. Second, we present a new visualization technique to add rainfall information. Third, we developed a scalable solution for water rendering to make it possible to investigate large-scale flooding scenarios. Finally, our system can be easily executed on network-connected display devices, hereby abstracting the used hardware—whether it be a single screen, a multi-screen setup, or a stereo device. In the following, we will give a short overview of these technical contributions.

Direct interaction with data points To facilitate discussions and support multi-user interaction, we designed a web-based solution for *collaborative* interaction that allows concurrent users to highlight areas or even modify data. The aforementioned changes in the simulation display (terrain lifting, soil changes, etc.) and annotations can be transferred directly to the virtual 3D LiDAR model. The on-the-spot visualization of concurrent changes on 3D topographic models of comparably sized datasets has not been demonstrated before within the scientific geospatial community. Our solution also allows highlighting of areas via common user interfaces, such as Lizard, Google Maps, or Open Street Maps to ensure a good acceptance of the system among the various users. Such possibilities can be helpful for discussions by directly visualizing the impact of the suggested solutions (Isenberg et al. 2013). The input applications can be executed on widely used smart devices (e.g., tablets and smartphones), making our algorithms accessible to a wide audience.

The modifications can manipulate any point property, such as color, but also height. The algorithm reads as an input keyhole markup language (KML) files, which are readily produced with the abovementioned software packages. These 2D polygonal definitions are shared among various users, who can then concurrently define modifications.

Technically, the approach modifies the LiDAR points on the fly by restricting the changes to the points that are currently visible to the observer in the 3D visualization software. Hereby, we avoid treating the entire data set, which would be too large to allow for interactive rates. Each LiDAR point is localized on the fly via a hierarchical structure build from the user-defined polygonal areas in the KML file (Kehl et al. 2013). The latter stores in each area the wanted attribute modification. This modification can then be applied to the localized LiDAR point, prior to being drawn on the screen.

Rainfall visualization Previous 3D visualization systems produced realistic imagery from simulation data, which improves understanding for practitioners. In this work, we further introduce a rainfall data visualization, which is particularly interesting for urban floods; dense construction (e.g., buildings, pavements, roads) and their disadvantageous infiltration properties lead to water accumulating on the pavement.

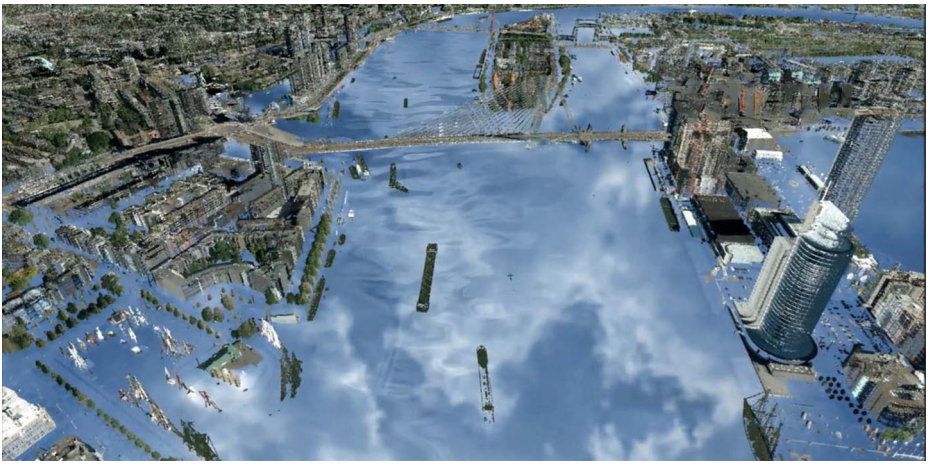


Fig. 2 Cloud reflection for a flood in the city of Rotterdam due to a hypothetical scenario of heavy rainfall (100 mm/h)

In order to naturally embed precipitation information in a realistic 3D visualization, we developed a method to render, dynamically update and animate clouds according to measurements provided by the Koninklijk Nederlands Meteorologisch Instituut (KNMI). Figure 2 shows an example of the reflection of these clouds for an urban flood scenario in the city of Rotterdam, but the cloud layer is also directly visible, e.g., in a top-down view. We render satellite imagery (Holleman et al. 2009) as a realistic cloud layer, positioned 10 km above the ground, that is streamed in real-time into our 3D visualization software.

Water rendering The rendering of large-scale water bodies is particularly challenging. No previous solution existed to directly render the results of hierarchical simulation processes (Stelling 2012b), despite the fact that such scenarios are very common in the investigation of large-area impacts.

The challenge is that the simulation relies on an approximation in the form of an adaptive quadtree, which respects water properties. These data sets, even in a hierarchical representation, are too large to be displayed entirely at real-time rates. Hence, an efficient display method is needed to simplify the representation. This adaptive scheme should be based on the viewing distance to add details only in the proximity to the observer, where they are needed.

The problem relates to terrain-rendering solutions, but previous work used a uniformly sampled height field (Losasso and Hoppe 2004; Baboud et al. 2011), while we need support for the hierarchical simulation results. Additionally, we want to support wave patterns to illustrate the underlying water properties, but previous work synthesized waves independently of any underlying simulation data (Ren and Zhou 2012) or did not allow for view-dependent simplifications (Kryachko 2004). These aspects are crucial for our large-scale data sets.

In our approach, we load and display only the simulation results that are currently in view. We use a hierarchical grid attached to the camera location that is very close to the observer and coarser in distance. As such, the grid resolution respects the distance to the observer. To transform this grid into a water representation, its vertices recover information from the underlying hierarchical flood simulation by querying the hierarchical structure based on their position. Depending on the stored water properties (presence, height, flow, and velocity), the vertices are displaced to represent the water levels and the recovered values are interpolated across each grid cell, which enables us to apply a texture-based wave-synthesis algorithm (van Hoesel 2011).

Display setups We designed a display algorithm that distributes the workload among network-interconnected rendering machines. Such a solution makes it easy to run our solution on various 2D displays, multi-display setups, as well as 2D and 3D stereo projector systems. This is an interesting aspect as different visualization setups are common (Marton et al. 2012; Reda et al. 2013; Kuchera-Morin et al. 2014).

Since particularly high-resolution results involve extensive computations in order to maintain interactivity at a high quality, our approach allows the involvement of multiple computers, which all contribute to the final images by sharing their results over the network. To illustrate the generality of this solution, we tested several different screen setups (screen walls of various screen arrangements, stereo devices, and cylindrical projection, as shown in Fig. 3). We also refer the interested readers to the accompanying video.

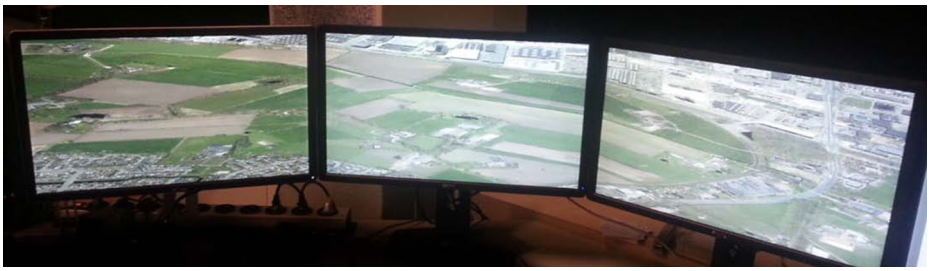


Fig. 3 Multi-screen setup, visualizing the city of Delft on a three-screen panorama

3 Flooding analysis

To test whether the two key features of our system (i.e., realistic visualization and interactivity) can contribute to a better accessibility of flood analysis tools for practitioners and support collaborative flood analyses with domain experts, we carried out two tests. The first was aimed at the usability of our system for individual practitioners who are no domain experts. The second was a use case to test whether our system could support collaborative flood analysis with practitioners and domain experts in a real-world application.

We gathered the data input for our system from the aerial LiDAR scan AHN2 (Actueel Hoogtebestand Nederland - www.ahn.nl) together with satellite imagery. The data was further augmented by object databases and soil maps provided by the Dutch Water Boards and conversion tables to transform land use into infiltration rates, as well as roughness, interception, permeability, and porosity values (CultuurtechnischeVereniging 1998). In common modeling practices, preferably a calibration and validation with real measured data is applied to assure the validity of the model outcomes. However, our case study was focused on testing the interface and user interaction and therefore a validation and calibration with real measured data was not necessary, although in this test, the outcomes had to be within a reasonable range of likelihood to be of use to the users.

3.1 User study

In this part, we evaluate the usefulness of our system for practitioners of flood management, who are usually no domain experts. To this end, we asked the participants to perform an analysis themselves and to comment on different types of visualizations obtained by our system.

3.1.1 Participants

Seven subjects participated in our study (5 males and 2 females, aged between 24 and 46—the mean age was 30). All had normal or corrected-to-normal vision. While they were not familiar with the system and goals, they all had experience in working with computers.

3.1.2 Methodology

The experiment took around 15 min per participant, including instructions, test, and subjective feedback. It consisted of two parts: (1) a comparison between a simulated flood hazard

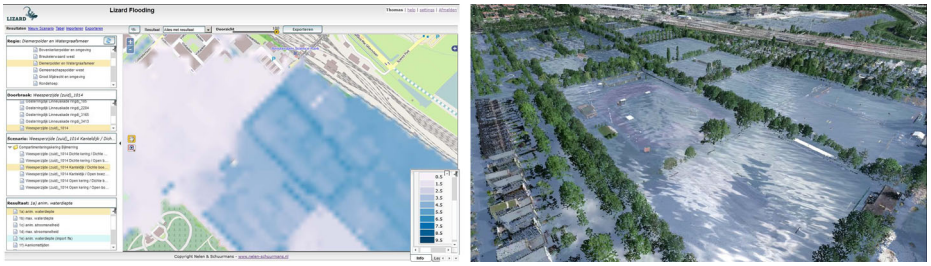


Fig. 4 A simulated flood hazard in the Watergraafsmeer area represented with an illustration directly from above (2D) and a helicopter view (3D).

represented with a illustration directly from above (2D) and a helicopter view (3D) (Fig. 4) and (2) an assignment in which the interactive simulation display was used.

For part 1 of the experiment, a flood was simulated in a study area in Amsterdam called Watergraafsmeer. The calculated maximum water depths of this simulation where projected in two ways. First, our 3D visualization system was applied, in which the user could make a realistic helicopter flight over the flooded area. Second, a conventional 2D visualization was applied, consisting of a digital map (Google), in which the maximum water depths where projected that could only be viewed directly from above (see Fig. 4). We showed both visualizations to each participant individually and asked them to respond to three closed questions and one open question:

1. Which visualization is best suitable to estimate damages to houses? [options: 2D, 3D, neutral]
2. Which visualization is best suitable to estimate loss of lives? [options: 2D, 3D, neutral]
3. Which visualization is best suitable to estimate whether evacuation is necessary? [options: 2D, 3D, neutral]
4. Does the 3D visualization have added value in respect to the 2D visualization and, and if so, what is this added value? [open question]

In part 2 of the experiment, the participants were asked to use the interaction possibilities to study the area Watergraafsmeer in Amsterdam. First, an introduction to our system was given to explain how the simulation display works. Second, the participants were given the opportunity to familiarize themselves with the simulation display. No explicit time limit was given for this. When they indicated that they understood how the simulation display had to be operated, they were asked to give an answer to the following question: which streets in the study area will be inaccessible for cars after a rain shower of 100 mm in 1 h? The time it took to answer this question was then measured.

3.1.3 Apparatus

We used an Intel Core i7 at 2.67 GHz processor, 6 GB of main memory, and an NVIDIA Quadro FX380 card with a 20-in. Samsung 2233RZ (120 H, 1680 × 1050 pixels) screen, and the participants used a classic three-button mouse device for interaction. For the 3D visualization, the main device uses an Intel Xeon 6 core hyperthreaded processor @ 3.2 GHz, 12 GB of main memory, and an NVIDIA GeForce GTX 680 with a common screen (60 Hz, 1920 × 1080 pixels) attached.

Table 1 The table presents the questions that where asked to each individual participant and, per question, the number of participants preferring a 2D/3D visualization

Question	2D Visualization	3D Visualization	Neutral
Which visualization is best suitable for the estimation of damages to houses?	2	5	0
Which visualization is best suitable for the estimation of loss of lives?	2	5	0
Which visualization is best suitable for the estimation whether evacuation is necessary?	3	3	1

3.1.4 Results user study

The responses to the closed question in part 1 of the test (i.e., the comparison between a 2D and 3D visualization of the same flooded area) are listed in the table below (Table 1)

The open question about what the added value of the 3D visualization was with respect to the 2D visualization was answered with the following statements:

- It makes it better possible to imagine the consequences of the flood.
- It enhances prediction of what a flood means for an area and helps to better empathize with the situation.
- It is more realistic and detailed. It is easier to interpret what a flood means for the area.
- It is more vivid and therefore better understandable.
- Less interpretation is required to estimate the consequences of the flood.
- It helps the user to better imagine how serious the flood is.
- It shows the consequences for the environment better.

The average time the participants needed to answer which roads would not be accessible for cars after a rain event of 100 mm was 6 min, with a minimum of 4 and a maximum of 7.

3.2 Case study

To test whether our system is accessible for both model experts and practitioners such that they can carry out flood analysis together in an interactive way, we organized a case study. This case study was organized together with the Province of North-Holland and the Waterboard Hollands Noorderkwartier in the Netherlands. Various stakeholders of the pilot area West-Friesland were invited to a workshop to discuss flood mitigation and adaptation measures within the multi-level safety approach.

The West-Friesland area is a large, flood prone area (781 km^2), located north of Amsterdam and lies approximately 3 m under sea level. It is protected by dikes from water in the IJsselmeer/Markermeer-lake. The area is inhabited by approximately 400,000 people.

3.2.1 Set-up case study

A broad workshop was organized to create a decision-making environment in which various stakeholders in the area were informed about the threats of floods and were involved in

investigating mitigation and adaptation measures in spatial planning and flood disaster management. During the workshop, the stakeholders were separated into groups to answer the central questions: "Which measures can be taken?" and "How can these measures be implemented?". Ideas of the stakeholders, resulting from their own backgrounds and perspectives on the threats of floods, could immediately be tested and discussed using our system. The system was also used to produce maps ahead of time. The results are the following four maps, each post-processed per flood scenario:

1. Flooded roads: highways, secondary, and urban roads
2. Flooded utility companies: divided into electricity, gas, and water
3. Flooded vulnerable objects; hospitals, day care centers, generated schools, and old age homes
4. Arrival times in order to provide information for planning evacuations.

Figure 5 shows an example of the arrival times of a simulated flood and Fig. 6 indicates the accessibility of roads.

3.2.2 Participants

Thirty-five stakeholders attended the workshop. This group existed of representatives of the province of North-Holland (official organizer of the workshop), the Waterboard, municipalities, agriculture, business, project developers, energy providers, health service, fire

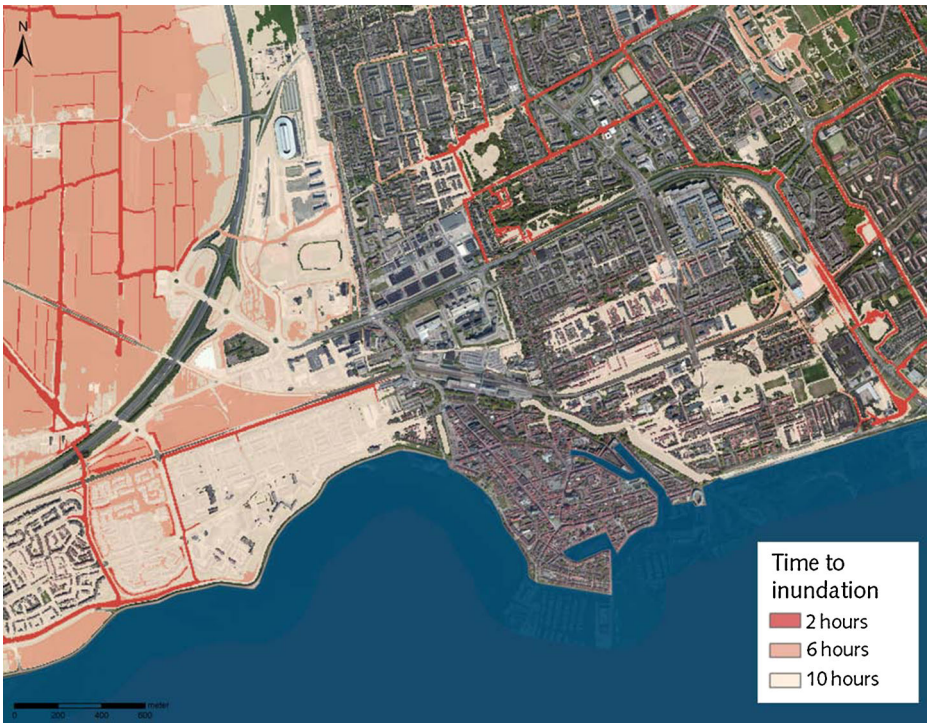


Fig. 5 Arrival times—an exemplary map showing flood arrival times in different zones, which can be used for city evacuation planning

Fig. 6 Accessibility of roads—another map resulting from the system that shows the mitigation of the flood (*left*) and the road access in the study area (*right*) accordingly. In the access map (*right*), *green paths* show usable road paths while *red-circled icons* symbolize access bottlenecks

department, and an insurance company. These stakeholders covered most of the parties involved in choosing mitigation measures. Apart from five domain experts from a consultancy company and the water board who where used to applying analysis tools, most of the other participants were not used to involvement in the application of analysis tools.

3.2.3 Results case study

The work session yielded a number of measures that were proposed by the participants and could be directly tested with our system. The measures were assigned to responsible stakeholders for further elaboration. An overview of these measures is listed in Table 2. One of the measures that emerged during the workshop, which is considered to be of high potential, was dividing the area in different components by dry dikes (Fig. 7). The evaluation of the use of our system in the workshop indicated that a combination of interaction with the system and the realistic visualization allows the users to quickly judge the effectiveness of

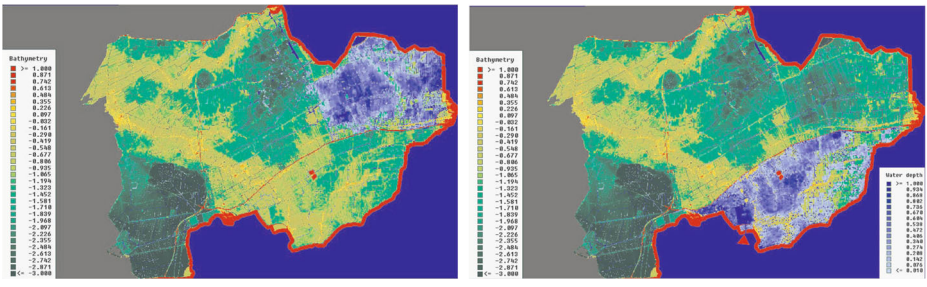


Fig. 7 One investigated measure for flood protection is shown in this figure. The illustration shows the effect of dividing the area of West-Friesland in a southern part and a northern part by a dry dike (*red line*) at both dam breach locations

Table 2 Flood mitigation and adaptation measures that were proposed by the participants of the work session and could be directly tested with our system

Stakeholder	Measure	Elaboration
Province	Include water safety in water regional planning	Use official strategic plan for spatial planning and raise and maintain regional roads in order to provide evacuation routes
Municipality	Incorporate water safety in building standards and regulations	Adapt official building standards
Water board	Create awareness and inform stakeholders on water safety	Apply the water system to reduce the consequences of floods, for example, by using compartments in discharge canals Supply flooding data and information on a non-expert level Apply the water system for flood reduction and practice
Suppliers of energy and water	Ensure drinking water during flooding events, by keeping the system under pressure (Electricity supply and communication systems tend to break down easily)	Keep pumping stations dry and assure emergency power supply Redirect mobile communication supply towards flooded area
Companies and entrepreneurs	Take private measures in case the level of protection ensured by the water board is not enough	Take local measures such as dikes around the property
Inhabitants	Take private measures to survive for a longer period in case of flooding	Prepare a survival kit
Emergency services (fire departments, police)	Switch from procedural scripts to scenario-related evacuations	Enhance evacuation scripts and the supply of information during calamities

The measures were assigned to responsible stakeholders for further elaboration

the solutions suggested in the various flood scenarios. The 3D visualization was considered a benefit over conventional 2D visualizations on maps. Users indicated that it was easier to estimate the impact of the calculated flood with the 3D visualization as they could relate the flood depths directly to familiar objects, such as trees, cars, and houses. With 2D visualizations, users first have to translate the different shades of blue into flood depths and then estimate the impact of the calculated flood. The positive impact on decision-making was attributed to the possibilities of testing the effectiveness of measures in a direct way. In the past, the effectiveness could only be evaluated by the intervention of model experts and was presented in a following work session. The post-processed map layers that illustrate additional information, such as damage, hindrance, and costs allocation, were deemed extremely useful for gaining additional insights beyond the physical values, such as water

depths or flow velocity. Furthermore, this representation addressed the legal responsibilities of the different stakeholders.

4 Discussion

The results of our user study show that our system is accessible and usable for users who are not domain experts like hydraulic engineers or model specialists. First, the 3D visualization helped most of the participants to better understand the consequences of a flood scenario in terms of damages, loss of life, and the urgency to evacuate in respect to the 2D visualization. In the open question about the added value of the 3D visualization over the 2D visualization, all participants agreed that the 3D visualization helped them to better imagine what a flood means for an area. Second, it was shown that these non-expert participants all were able to use our interactive simulation display. Moreover, they were able to carry out an expert analysis about the availability of roads after a heavy rain event within minutes. With conventional models, which have specialist user interfaces and require hours of computation time, this would not have been possible.

The case study showed that our system was usable for collaborative application by practitioners and domain experts. The simulation display was effectively used on the level of at least three aspects. First, it enabled practitioners from different backgrounds to understand what floods meant for their specific field of profession. Second, it helped these practitioners to contribute in suggesting suitable solutions, following the multi-level safety approach. Third, the simulation display helped the practitioners to retrieve direct feedback on the effectiveness of their suggested solutions.

Our results show that our system is accessible for practitioners of flood management such that they can carry out flood analysis together with domain experts in interactive work sessions. However, one should be careful to draw general conclusions about the appreciations of our system. In our case, the simulation display and 3D visualization were mainly used for a general exploration of solutions which on the short term have no direct financial or political consequences. The impact of the decisions and the time pressure for making these decisions were therefore low. It is expected that our system will be used differently when time pressure and consequences are high. Experiences show that decision-makers in these circumstances highly value the accuracy of the outcomes of a model (Brugnach et al. 2007).

We expect that interactive modeling can contribute to a better understanding of uncertainties in model outputs among decision-makers as they can directly examine if current data were used in the model setup. They can also see how suggested solutions are translated into the model and therefore better understand the scope of the outcomes. Still, a calibration and validation with real measured data, as applied in common model practices, is always advisable. However, in environmental problems related to climate changes, the future is unpredictable and the nature of the relationship between processes is sometimes unknown (Leedal et al. 2010). In these cases, different model concepts or variation in input data in ensemble calculations can be considered (Renner et al. 2009; Walker et al. 2003). Methods to cope with model uncertainty, such as the use of real measured data or ensemble calculations, are all applicable with our system.

Regarding the technical aspects of our studies, we report that our system is efficient enough to be executed on a standard desktop PC, as reported in Section 3.1.3. For all scenarios, we could ensure a framerate of roughly 20 frames per second, which leads to a good tradeoff between speed and accuracy. Figure 8 shows a few images from a flythrough with

Fig. 8 Our system is interactive and maintains a stable framerate. Here, we show an evolving flood during a real-time flythrough on a standard desktop computer

an evolving flood. The interaction component was executed on a tablet but would run on most web-capable devices, which is a big advantage as it makes a large part of our system accessible anywhere and to anyone.

5 Conclusions

In this paper, we presented our system for analyzing flooding scenarios. We showed that, despite the complexity of the involved models and the size of the involved data sets, our system is accessible for practitioners of flood management such that they can carry out flood analysis together with domain experts in interactive work sessions. In particular, the realistic 3D visualization helps practitioners, who are no domain experts, to better estimate the scale as well as the impact of a flood. The simulation display allows them to interact with the system and to explore complex flooding scenarios. Both results have been demonstrated in a user study and a case study, showing that our system represents an important step ahead towards the closer involvement of practitioners in the analysis driving the decision-making process for developing mitigation and adaption strategies for the threats of floods.

The accessibility of our system can also be underlined by the fact that a modified version was recently installed in two museums in The Netherlands, namely the Delft Science Center, as well as the Watersnoodmuseum in Zeeland. The two installations rely on our 3D visualization system to illustrate the impact of the 1953 flooding, which led to an enormous destruction of large parts of the Netherlands. Our solution gives people a direct access to flood-related information and offers insights into decision-making for flood mitigation and adaption strategies.

Acknowledgments The necessary software development, testing procedures, and use cases were collectively supported and funded by the waterboards Delfland and Hollands Noorderkwartier and by the national research program of "Knowledge for Climate"[1]. "Knowledge for Climate" furthermore offered a communication base for sharing insights and arranged collaborations with other research teams. Some researchers were funded by the European FP7 Harvest4D project.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

[1]Kennis voor Klimaat - http://kennisvoorklimaat.klimaatonderzoeknederland.nl/

References

Al-Sabhan W, Mulligan M, Blackburn G (2003) A real-time hydrological model for flood prediction using gis and the www. Comput Environ Urban Syst 27:9–32

Apel H, Thieken AH, Merz B (2004) Flood risk assessment and associated uncertainty. Nat Hazards Earth Syst Sci 4:295–308

Baboud L, Eisemann E, Seidel H-P (2011) Precomputed safety shapes for efficient and accurate height-field rendering. IEEE Trans Vis Comput Graph 99. PrePrints

Barros V, Dokken D, Mach K, Mastrandrea M, Bilir T, Chatterjee M, Ebi K, Estrada Y, Genova R, Girma B, Kissel E, Levy A, MacCracken S, Mastrandrea P, White L (2014) Climate change 2014: impacts, adaptation, and vulnerability. Part A: Global and sectoral aspects. Contribution of working group II to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change

Bates P, De Roo A (2000) A simple raster-based model for flood inundation simulation. J Hydrol 236(1):54–77

Bates PD, Roo A. PJD (2000) A simple raster-based model for flood inundation simulation. J Hydrol 236:54–77

Brugnach M, A T, Keil F, de Lange W (2007) Uncertainty matters: computer models at the science-policy interface. Water Resour Manag 21:1075–1090

Casulli V, Stelling GS (2013) A semi-implicit numerical model for urban drainage systems. International Journal for Numerical Methods in Fluids

CultuurtechnischeVereniging (1998) Cultuurtechnisch vademecum

Dachsbacher C, Vogelgsang C, Stamminger M (2003) Sequential point trees. In: ACM SIGGRAPH 2003 Papers, SIGGRAPH '03, pages 657–662, New York, NY, USA. ACM

EU (2005) DIRECTIVE 2007/60/EC on the assessment and management of flood risks

Faulkner H, Parker D, Green C, Beven K (2007) Developing a translational discourse to communicate uncertainty in flood risk between science and the practitioner. Ambio 36:692–703

Haan Gd (2009) Scalable visualization of massive point clouds. Nederlandse Commissie voor Geodesie KNAW 49:59–67

Haan Gd (2010) Interactive visualization of massive aerial LiDAR point clouds. In: European LiDAR Mapping Forum

Holleman I, Beekhuis H, Roozekrans H (2009) Update of KNMI HDF5 data format specification v3.6. Intern report, Koninklijk Nederlands Meteorologisch Instituut (KNMI), De Bilt

Isenberg P, Isenberg T, Hesselmann T, Lee B, von Zadow U, Tang A (2013) Data visualization on interactive surfaces: a research agenda. IEEE Comput Graph Appl 33(2):16–24

Janssen JAEB, Krol MS, Schielen RMJ, Hoekstra AY (2010) The effect of modelling expert knowledge and uncertainty on multicriteria decision making: a river management case study. Environ Sci Pol 13:229–238

Kabat P, Fresco LO, Stive MJ, Veerman CP, van Alphen JS, Parmet BW, Hazeleger W, Katsman CA (2009) Dutch coasts in transition. Natl Geogr 2:450–452

Kehl C, de Haan G (2012) Interactive simulation and visualisation of realistic flooding scenarios. In: Intelligent Systems for Crisis Management

Kehl C, Tutenel T, Eisemann E (2013) Smooth, interactive rendering techniques on large-scale, geospatial data in flood visualisation. In: Information and Communication Technologie (ICT) Open

Kryachko Y (2004) GPU Gems 2, chapter Chapter 18: Using Vertex Texture Displacement for Realistic Water Rendering. Addison-Wesley

Kuchera-Morin J, Wright M, Wakefield G, Roberts C, Adderton D, Sajadi B, Hòllerer T, Majumder A (2014) Immersive full-surround multi-user system design. Comput Graph 40(0):10–21

Leedal D, Neal J, Beven K, Young P, Bates P (2010) Visualization approaches for communicating real-time flood forecasting level and inundation information. J Flood Risk Manage 3(2):140–150

Leskens JG, Brugnach M, Hoekstra A (2014a) Application of an interactive water simulation model in urban water management, a case study in amsterdam. Water Sci Technol 70:1729–1739

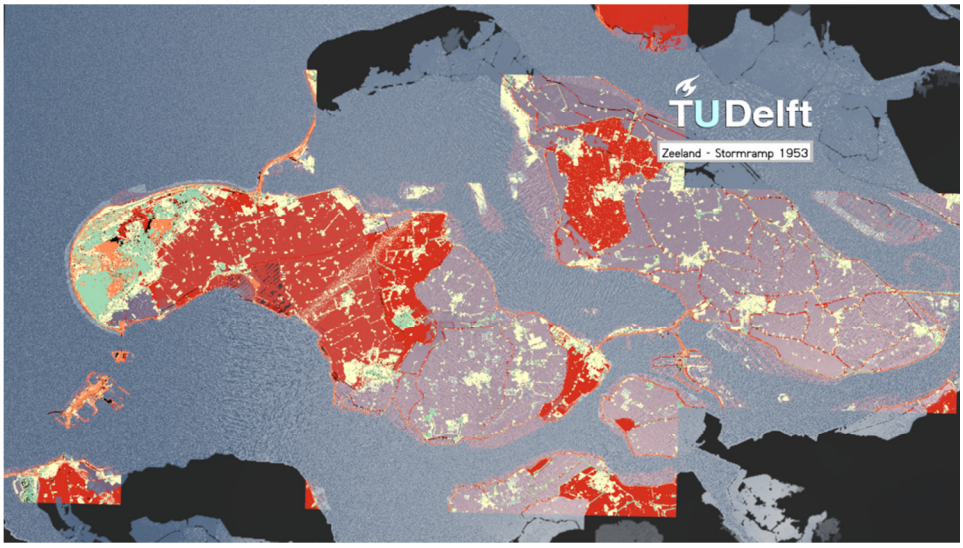
Leskens JG, Brugnach M, Hoekstra A, Schuurmans W (2014b) Why are decisions in flood disaster management so poorly supported by information from flood models Environ Model Softw 53:53–61

Losasso F, Hoppe H (2004) Geometry clipmaps: terrain rendering using nested regular grids. ACM Trans Graph 23(3):769–776

Marton F, Agus M, Gobbetti E, Pintore G, Balsa Rodriguez M (2012) Natural exploration of 3D massive models on large-scale light field displays using the fox proximal navigation technique. Comput Graph 36(8):893–903

McGranahan G, Balk D, Anderson B (2007) The rising tide: assessing the risks of climate change and human settlements in low elevation coastal zones. Environ Urban 19:17–37

Moel HD, Aerts J (2011) Effect of uncertainty in land use, damage models and inundation depth on flood damage estimates. Nat Hazards 58:407–425
Morss RE, Wilhelmi O, Downton MW, Gruntfest E (2005) Flood risk, uncertainty, and scientific information for decision making: lessons from an interdisciplinary project. Bull Am Meteorol Soc 86:1593–1601
Nóbrega R, Sabino A, Rodrigues A, Correia N (2008) Flood emergency interaction and visualization system. Springer
Reda K, Febretti A, Knoll A, Aurisano J, Leigh J, Johnson A, Papka M, Hereld M (2013) Visualizing large, heterogeneous data in hybrid-reality environments. IEEE Comput Graph Appl 33(4):38–48
Ren X, Zhou X (2012) Gpu-accelerated large-scale water surface simulation. In: Int Proc Comput Sci Inf Technol, vol 50, pp 187–194
Renner M, Werner M, Rademacher S, Sprokkereef E (2009) Verification of ensemble flow forecasts for the river rhine. J Hydrol 376(3):463–475
Rusinkiewicz S, Levoy M (2000) Qsplat: A multiresolution point rendering system for large meshes. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '00 pages 343–352, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co
Stelling GS (2012a) Quadtree flood simulations with sub-grid dems. Water Manage 165:1–14
Stelling GS (2012b) Quadtree flood simulations with sub-grid dems. Water Manage 165:1–14
STOWA (2008) Richtlijn - Normering Keringen langs Regionale Rivieren (Guideline Standardization of Safety Levels of Regional Levees)
Timmerman JG, Beinat E, Termeer CJAM, Cofino WP (2010) A methodology to bridge the water information gap. Water Sci Technol 62:2419–2426
van Hoesel F (2011) Tiled directional flow. Talk at SIGGRAPH
Voinov A, Bousquet F (2010) Modelling with stakeholders. Environ Model Softw 25:1268–1281
Walker WE, Harremoës P, Rotmans J, van der Sluijs JP, van Asselt MB, Janssen P, Krayer von Krauss MP (2003) Defining uncertainty: a conceptual basis for uncertainty management in model-based decision support. Integr assess 4(1):5–17
WHO (2013) Annual report
Wood M, Kovacs D, Bostrom A, Bridges T, Linkov I (2012) Flood risk management: US army corps of engineers and layperson perceptions. Risk Anal 32:1349–1368



Interactive visual analysis of flood scenarios using large-scale LiDAR point clouds

Tim Tutenel, Christian Kehl, Elmar Eisemann – Delft Univeristy of Technology, Netherlands

Abstract Geospatial World Forum 2013

The visualisation of large-scale geospatial data is a demanding challenge that finds applications in many fields, including climatology and hydrology. In this work, we are focusing on the visual analysis of flooding scenarios. We propose a novel solution that is able to visualise coloured LiDAR data of several hundred square kilometres including high-resolution flood simulation results.

One key contribution is a level-of-detail rendering algorithm that enables us to handle Terabytes of data interactively. Furthermore, we present a novel algorithm to annotate (colour entire regions or highlight routes), or modify (i.e. clip, cut, raise/lower) the unordered LiDAR point set on-the-fly with flexible 2D geographic metadata (i.e., polygons, paths and landmarks). The principles of our technique are to make use of hierarchical structures on the metadata and a new quadtree-based GPU traversal algorithm.

We examined three test cases and show that combining the flood simulation with different topographic shading techniques facilitates drawing conclusions about water flow, security measures, and evacuation planning.

ChromoStereoscopic Rendering for Trichromatic Displays

Leïla Schemali[1,2] Elmar Eisemann[3]
[1]Telecom ParisTech CNRS LTCI [2]XtremViz [3]Delft University of Technology

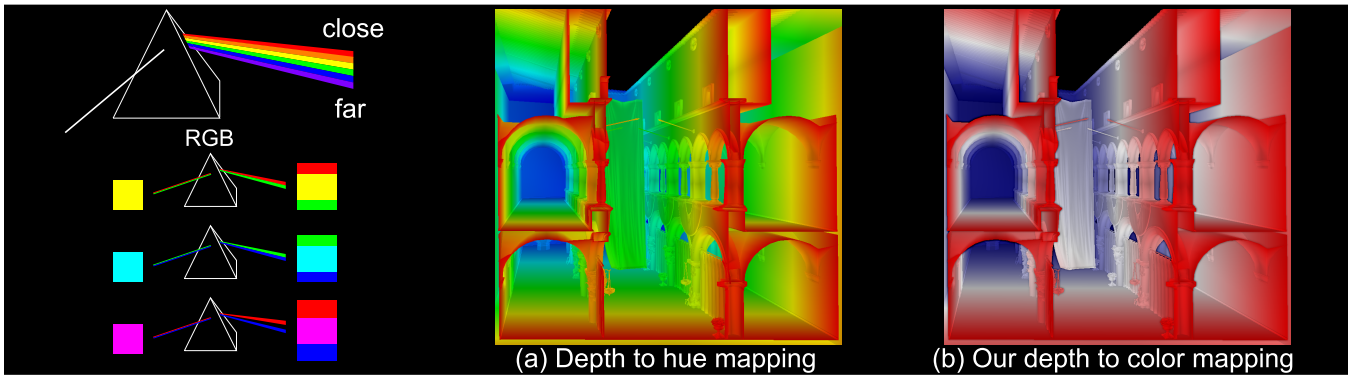


Figure 1: *ChromaDepth®glasses act like a prism that disperses incoming light and induces a differing depth perception for different light wavelengths. As most displays are limited to mixing three primaries (RGB), the depth effect can be significantly reduced, when using the usual mapping of depth to hue. Our red to white to blue mapping and shading cues achieve a significant improvement.*

Abstract

The chromostereopsis phenomenom leads to a differing depth perception of different color hues, e.g., red is perceived slightly in front of blue. In chromostereoscopic rendering 2D images are produced that encode depth in color. While the natural chromostereopsis of our human visual system is rather low, it can be enhanced via ChromaDepth®glasses, which induce chromatic aberrations in one eye by refracting light of different wavelengths differently, hereby offsetting the projected position slightly in one eye. Although, it might seem natural to map depth linearly to hue, which was also the basis of previous solutions, we demonstrate that such a mapping reduces the stereoscopic effect when using standard trichromatic displays or printing systems. We propose an algorithm, which enables an improved stereoscopic experience with reduced artifacts.

CR Categories: I.3.3 [Computer Graphics]: Three-Dimensional Graphics and Realism—Display Algorithms;

Keywords: ChromoStereoscopy;Stereoscopic rendering

1 Introduction

Artists have made use of color to encode and induce depth perception for centuries [Livingstone 2002] - a well-known example is the application of aerial perspective. In general, we tend to perceive colder colors as more distant than warm tones. One possible explanation is a subtle natural chromatic aberration induced by the lens of our visual system [Einthoven 1885]. Usually, this effect is rather weak, but strong enough to find applications in visualization systems [Chu et al. 2008].

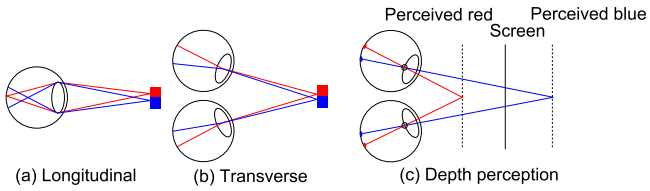


Figure 2: ***Chromostereopsis*** *can be due to: (a) longitunal chromatic aberration, focus of blue shifts forward with respect to red, or (b) transverse chromatic aberration, blue shifts further toward the nasal part of the retina than red. (c) Shift in position leads to a depth impression.*

This chromostereopsis phenomenon can be enhanced using specialized glasses, e.g., ChromaDepth®glasses. These glasses typically consist of a standard window glass for the right eye and a special prism-like holographic film for the other, which induces an increased chromatic aberration. Due to the positional shift of different wavelengths in one eye, but not the other, disparity between the image pairs is produced that is perceived as depth. For example, a red object will be shifted to the right for the left eye, hereby, creating the illusion of this object being closer to the observer.

Based on the wavelength, light is consistently refracted to induce a visible shift. Nonetheless, an LCD screen and printer can usually not synthesize arbitrary wavelengths, but only combine a limited set of base colors to produce the illusion of various hues. For screens, subpixel information in form of a red, green, and blue channel are combined. For chromostereopsis, this limitation has a significant impact and can lead to artifacts. Imagine the color yellow (actually represented via red and green), which maps to two distinct positions. In this paper, we will take these observations into account to derive more effective chromostereopsis for LCD screens.

Our solution achieves an increased depth perception and reduces visual artifacts. It improves the quality of chroma-based stereo, which is interesting for several reasons. First, images are easy to produce with our solution. Second, the necessary stereo equipment is cheap. Third, the result is naturally backwards-compatible; an

observer without glasses perceives a clear and crisp image and only lacks the enhanced depth perception. Also, the image builds upon a warm-cold contrast, which makes it a suitable representation for visualization purposes [Gooch et al. 1998; Rheingans 1999].

Specifically, our contributions are: an analysis of chromostereoscopy for RGB displays; a fast and easy-to-implement method for chromostereoscopy; enhancement techniques to improve depth perception; an informal user study to evaluate the quality of the results.

2 Related Work

Einthoven's theory [1885] first described chromostereopsis induced by our eyes. On a black background, we perceive red in front of the image plane and blue behind it. The phenomenon was attributed to a different refraction index for each color of the light spectrum when a ray enters the eye. The blue projection on the retina is shifted more to the nasal part (Fig. 2(b) and (c)), known as the transverse chromatic aberration. This explanation is generally preferred over the longitudinal chromatic-aberration theory (Fig. 2(a)), stating that the focus point in the eye appears closer for blue than for red [Simonet and Campbell 1990; Ye et al. 1991; Faubert 1994].

To exploit and amplify the natural chromostereopsis effect, special glasses have been introduced. The Chomatek Corporation developed the ChromaDepth®glasses, which are composed of superchromatic prisms. The visible light is refracted with a refraction index depending on the light wavelength, enhancing the color-depth effect [Steenblick 1987; Sicking et al. 1995; Ucke 1999]. The lower wavelength, violet, is refracted toward the nasal part of the retina while higher wavelengths, red, are redirected towards the temporal part. Hereby, a disparity between the left and right retinal images is induced, which is interpreted by the human visual system as a difference in depth (Fig. 2 (c)). We can theoretically create a depth ramp by mapping depth linearly to the visible light spectrum between red (close) and violet (distant). The background of the image should be black, as, otherwise, the background color will also be refracted and interferes at object boundaries (Fig. 3). Initially, two holographic films (one for each eye) were used, but this process lead to a reduced sharpness [Ucke 1999].

The chromostereoscopy along with the ChromaDepth®glasses has been used extensively in various areas of computer graphics. For scientific visualization [Bailey and Clark 1998], for cartographic images [Toutin 1997; Petrie et al. 2001], complex graph visualization [Wallisch et al. 2002], solar physics simulation data [Verwichte and Galsgaard 1998], and even marine environment [Hamid 2012].

All these previous sources suggest a linear depth-to-hue mapping (red for close, green for middle, blue for distant elements) to produce chromostereoscopic images, referenced hereafter as hue mapping. However, on a trichromatic (RGB) screen, the so-defined depth is not optimal, appears non linear, and color artifacts on the boundary can be very significant, e.g., for yellow (Fig. 3(a)).

For angiography visualization, Ropinski *et al.* [2006] proposed a modified chromostereoscopic rendering using red, magenta and blue, from the closest to the farthest, on a black background (Fig. 3(b)). Unfortunately, magenta tends to be perceived at the same depth as red instead of between red and blue [Bailey et al. 2006]. We will analyze the reasons in this paper and propose a new color scheme that better matches the expectations.

Additionally, a chromostereopsis rendering algorithm, using non-photorealistic techniques, such as strokes and silhouette drawing, has been developped by Chu *et al.* [2008] for vascular visualization. They draw strokes parallel to the vessel section, varying their color from red to green to blue, on a black background. Further, they add a white silhouette, which is not a good choice, as it is perceived at the screen depth, hereby, contradicting the stroke depth. In this work, we also optionally add shading cues at silhouettes to improve depth perception and show their benefit.

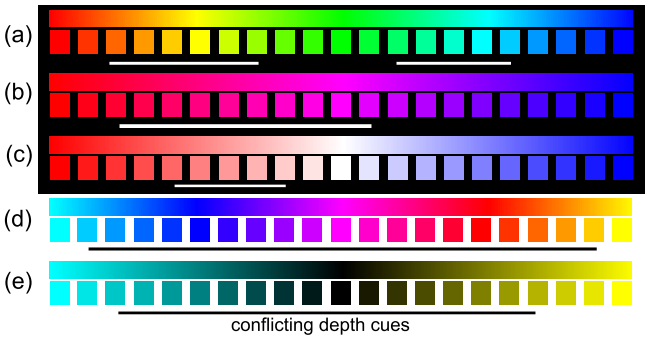
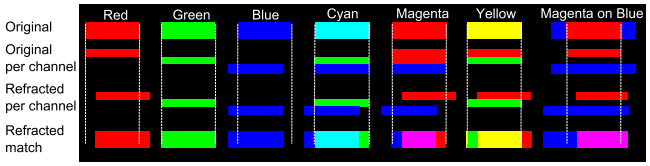


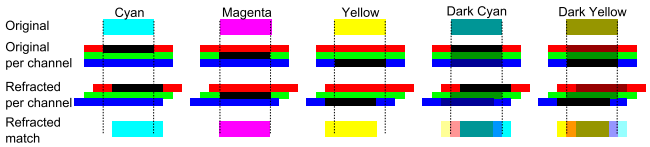
Figure 3: *Colors leading to a depth ramp; a red-to-blue on a black and cyan-to-yellow ramp on a white background. Underlined colors lead to conflicting depth cues on trichromatic displays.*

3 ChromoStereoscopy with RGB Displays

Most displays and prints are trichromatic, meaning that three primitive colors are mixed. Consequently, there are also only three distinct shifts associated to these color primitives to induce a depth illusion (Fig. 4). The hue mapping between red and blue is only a weak approximation of different spectra, although it is true that at least blue shifts to the left, red to the right and green is mostly invariant. A red (green/blue) square on a black background will, thus, be seen close (at screen distance/far away). Nonetheless, this observation does not hold for all colors; yellow will be perceived at screen distance and show color artifacts on its boundaries; the red and green channel being independently refracted by the glasses, a similar observation can be made for cyan.



(a) Black background



(b) White background

Figure 4: *Color perception on a black or white background looking through the left holographic foil of ChromaDepth®glasses. (a) Red shifts right, blue shifts left, green does not move. Colors combining various channels show fringing. (b) The white background is decomposed in blue, green and red and mixes with the color patches. Darkened yellow and cyan do no longer show a chromostereoscopic effect.*

The use of different background colors can have a strong impact on the visible depth position of a patch. If background and patch channels share information (e.g., red object on a yellow background),

there is necessarily an interaction between the two. This cross talk can be strong enough to hinder the human visual system from establishing correspondences between both views, leading to an annihilation of the entire stereoscopic effect. E.g., a cyan patch on a black background will be perceived at screen distance because cyan produces a green and blue copy. The visual system has a weakness for blue and will tend to match cyan with green, whose retinal projection is not influenced by the holographic foil. A cyan patch on a white background corresponds to the inverted situation of red on a black background. In other words, it only differs in the red channel with respect to the background, hence, it will be shifted and thus be seen in front of the white background. Nonetheless, if one were to use a dark cyan patch, the matching will fail, as it induces several differently-colored borders that hinder the stereoscopic perception. On the one side a blue/cyan, on the other a pink/yellow border. In both cases, the strong differences in the green/red channel avoid a perfect match. Further, the red and blue boundary, even if matched, lie at the initial position, hence, the chromostereoscopic effect is always annihilated (Fig. 4(b)).

The screen physical sub-pixel arrangement (horizontal/vertical) does not modify the way the color fringes appear on the vertical edges of an object whose color is composed of several primaries. We observed that for a viewing distance of 60cm, red (blue) is shifted to the right (left) by an amount of around 6 pixels (1.5mm)

4 Our Depth-to-Color Mapping

To define our rendering algorithm, we will build upon the previous observations to improve upon the hue mapping. In order to avoid most interference, we choose a white or black background. In this case, the color channels will always represent extremes with respect to the object colors. For the moment, we will first investigate the case of a black background.

To cover the entire possible depth range, a red-to-blue mapping, via magenta, might seem an appropriate choice. In practice, the solution does not perform well because the red and blue channels are consistently warped. The red channel moves in one direction, the blue channel in the other, both with constant distance. Hence, it looks like two semi-transparent foils, a red one in front of a blue one. The scene mostly appears like a two-layer depth representation of transparent objects, than a consistent 3D scene (Fig. 4(a),12(c)).

In order to avoid the layered impression, one could consider a transition from red to blue via green, which is perceived at screen distance. Unfortunately, the introduction of a green channel, while fading out the red channel can lead to matching conflicts. Before most, our human visual system matches luminance [Heckmann and Schor 1989; Legge and Yuanchao 1989; Didyk et al. 2012] and the fact that the eye is slightly more sensitive to green seems to result in a mismatch when showing red and green at the same time. This observation seems to match the previous remark regarding yellow being a bad choice because the borders decompose into red and green bands. In particular, it is evident that the hue mapping, which includes yellow, as well as cyan, thus results in artifacts and mismatches, which induce a non-monotonic depth repartition. In Fig. 3, we show various mappings for comparisons.

Our choice is to rely on a red to white gradient for nearby objects up to the screen distance. Fading in the green and the blue channel simultaneously to produce white, without reducing the red channel, improves the matching drastically. First, white will be perceived at the screen distance, just like green, but it shows actual boundaries. The inner boundaries are yellow and cyan, which are close enough to white to make a matching possible and result in the screen distance. Additionally, white shows an outer red and blue boundary, but both colors are too different to be matched. When fading green and blue, red will be dominated and the matching process will continuously shift away from the red location towards the original position (Fig. 5).

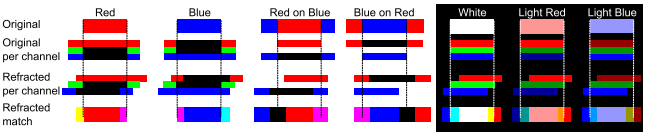


Figure 5: *Possible interactions between red, blue and white.*

We believe the reason why this shift is continuous can be explained via the observation that cross fades can induce motion [Kemelmacher-Shlizerman et al. 2011]. It was shown that blending two Gaussians can result in a perceived continuous shift. As the holographic foil leads to a certain blur and the matching is predominantly based on luminance, we are in a similar situation. The induced motion being mostly linear, we can assume that the depth perception will be as well.

To achieve a transition from the screen to distances further beyond, we employ a white to blue gradient. The same argumentation as before holds to explain the continuous shift. Nonetheless, it is true that we initially expected the second transition to be very non-linear due to the weak blue perception of the human visual system. In practice, this effect seems negligible and the process is governed rather by the absence of red and green. We, thus, compute the color by mapping the depth interval as follows. From near to half the interval, we attribute a hue of 0 (red), the other half is colored with a hue of 240 (blue). The transition towards white is obtained by linearly interpolating the saturation, which is set to 1 for the extremes (near and far) and 0 for the middle.

When using a white background, we need to invert our color scheme. The mapping would then result in cyan over black to yellow. In theory, one could use a cyan/white/yellow scheme, but then the differentiation from the background is impossible. This color scheme works well in practice and performs better than the hue mapping. However, using an inverted scheme still has a significant downside with respect to the red/white/blue color scheme on a black background; it is possible to modulate the brightness levels of the red/white/blue scheme without significantly modifying the depth impression. Given that the inverted scheme makes use of black, such a modification would be impossible. The possibility to change the brightness levels without impacting depth perception is very useful when adding shading cues and we will explore this possibility in the next section.

5 Additional Cues

In this section, we present three ways to modify the standard mapping from depth to color to increase the depth illusion.

5.1 Shading

Besides binocular stereo, there are many other depth cues that help our visual system in understanding a scene [Palmer 1999]. One important aspect is shading. The use of the red/white/blue color mapping enables us to modulate the brightness of these colors according to the shading of the scene. An example, using Lambertian shading is shown in Fig. 6. Adding shading cues enhances the binocular stereo effect because feature matching during the fusion process is simplified. While the light source can be placed in any location, we found that a standard lighting from the right/left upper quadrant with respect to the view is effective.

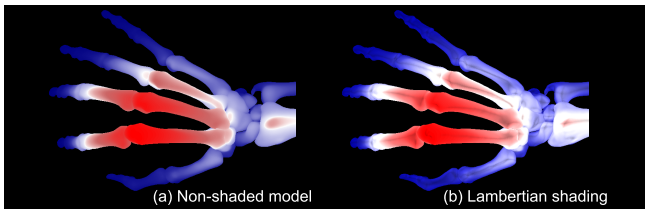


Figure 6: *Effect of shading on shape and depth perception.*

5.2 Unsharp Masking

Besides shading, we introduce an additional modification that enhances silhouettes to improve depth perception. One important observation is that when not considering patches but smoothly varying color transitions, the color scheme is less crucial. The reason is that in these situations the subtle differences of neighboring pixel colors are not easily discernable. Consequently, the image observed via the holographic foil does not significantly differ, except at region boundaries. Hence, depth discontinuities are the most important depth cues, which also matches findings regarding depth perception that can be induced from discontinuities in the scene [Didyk et al. 2011; Brookes and Stevens 1989; Anstis et al. 1978].

To increase depth perception at discontinuities, we make use of depth unsharp masking [Luft et al. 2006]. Hereby, we can limit the interactions between different colors and increase the impact of depth discontinuities. Adding darkened edges at discontinuities keeps boundary colors from mixing with the colors of more-distant geometry. The edges then project on a black background, leading to improved stereo matching.

Depth unsharp masking has been introduced to enhance silhouettes by introducing a special gradient, a so-called Cornsweet profile [Anstis et al. 1978], at silhouettes. The latter is produced by summing the color image and a scaled unsharp mask, which is defined by the difference between the original depth map and its Gaussian-filtered version. Usually, the scale factor is a variable that defines the strength of the effect. In our case, we keep this value equal to one. An additional change is necessary to avoid modifying the appearance of the occluding surface, but to only darken the underlying surface. As the unsharp mask contains negative values for the farther and positive values for the nearer surface next to a depth edge, we can achieve this goal easily by keeping only the negative values and setting all others to zero (Fig. 7). In this way, depth discontinuities, but also small features can be pronounced and the effective depth perception is increased.

The only remaining parameter of the unsharp mask is the variance of the Gaussian blur kernel. In order to avoid an overlap between the shifted projection and the surface below. When analyzing the shift, we saw that at 10 cm viewing distance from the screen, the shift equals roughly one pixel (0.25mm), as the shift is based on a refraction by the holographic foil, the displacement increases linearly with the distance to the screen. We can thus keep the blur kernel proportional to the shift in order to achieve an adequate edge darkening (Fig. 7).

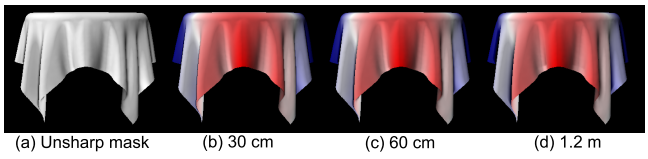


Figure 7: ***Depth unsharp masking****: The depth discontinuities are outlined by darkening the occluded surface along the depth edge. The darkening is greater when the viewing distance increases.*

5.3 Depth Warping

In recent years, various papers have focused on the question of how to map the depth of a 3D scene appropriately for stereo visualization. Besides automatic [Oskam et al. 2011] and hand-defined [Wang and Sawchuk 2008] adaptations, also manipulations based on perception start to be better understood [Didyk et al. 2011; Didyk et al. 2012]. In our context, it is also possible to warp the depth values prior to applying the color conversion. One meaningful warping is a histogram adjustment to spread the depth values most efficiently over the whole color range. Alternatively, the median can be used to separate the scene in the area in front and behind the screen (Fig. 8).

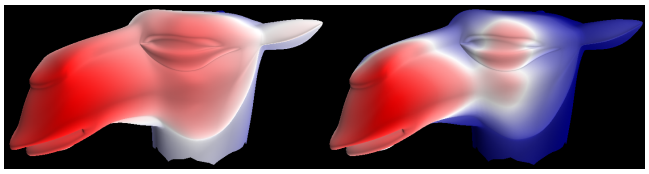


Figure 8: *Left: linear depth mapping, Right: equal number of pixels between red/white, and white/blue for better depth-range usage.*

6 Evaluation

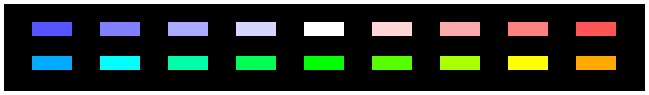


Figure 9: *Palette presented to the user study participants.*

We conducted an informal user study with 11 participants (age: 23–33, mean: 28, 8 males, 3 females) with normal or corrected-to-normal vision and relied on a Samsung 2233RZ screen that was observed from 60 cm distance using the ChromaDepth®glasses. We showed several stimuli and collected the participants feedback after each one. All participants were allowed to take breaks and there was no time constraints on the evaluation.

In a first step, we used a color palette (Fig. 9). The patches vary from light red to light blue on the first row and from yellow to cyan on the second row, on a black background. This test is used to compare the previous hue-based solution with our red/white/blue color mapping for *in-between* depths. For both examples, we asked the participants if they were able to perceive a change in depth and if so, how the patches differ in depth. Nine out of eleven participants saw light red slightly in front of the screen and light blue slightly behind it. The other two saw all the patches at the same depth. In contrast, for the hue mapping, all participants saw all patches at the same depth. This illustrates our previous claims regarding green, yellow, and cyan on a black background to be perceived at screen distance.

In a second step, we showed several chromosterescopic examples to illustrate some of the application areas mentioned in the previous work section; a 3D scene (Fig. 1), blood vessels (Fig. 10), and a point cloud (Fig. 11).

The 3D scene (Fig. 1) was shown with our and the hue color mapping. Asked for preference, all participants chose our over the

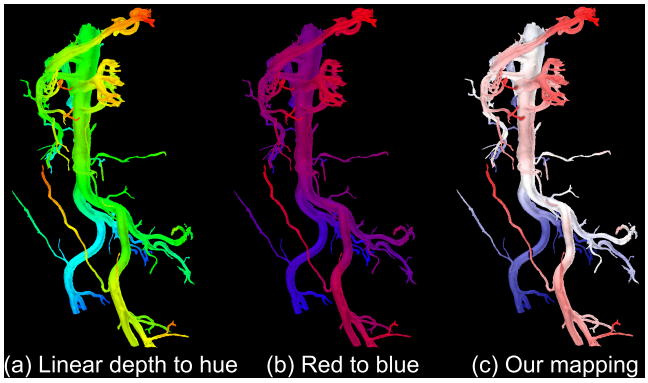


Figure 10: *Blood vessels of the urologic system. Our mapping leads to a natural and strong stereo perception, while a* red magenta blue *mapping leads to abrupt discontinuities.*

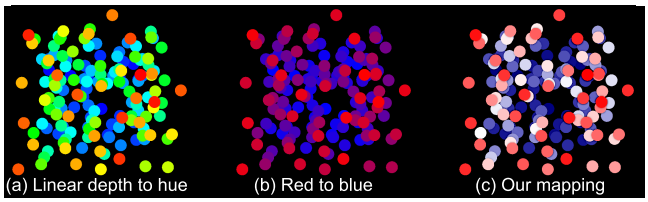


Figure 11: *Point cloud in chromostereoscopy. Our mapping produces a larger depth range than a hue mappping.*

red/green/blue color mapping. Further, when asked about the wider depth range, ten out of eleven chose our over the hue mapping.

For the blood vessel (Fig. 10), we used our, the hue-based and the red/magenta/blue mapping, as it was actually introduced for a similar purpose [Ropinski et al. 2006]. We asked the participants to identify the part of the model that appeared closest to the point of view. All participants identified the red/magenta/blue image to provide the widest depth range, but all added that they could actually perceive only 2 depth layers: red and magenta in front of the screen and blue behind, making it impossible to determine which part of the model was the closest. For our color mapping, all the participants identified the upper-right part of the model as being the closest, which also coincided with the actual 3D configuration. In contrast, for the hue mapping, the answers differed drastically; upper right (4), the upper left (3), the bottom right (1) and the middle left (3). With this mapping, the attention of the viewer is drawn toward the yellow parts of the model [Rheingans 1999].

Finally, for the point-cloud example (Fig. 11), we asked how many distinct depth levels they could clearly differentiate. We showed them the hue, our, and a red/magenta/blue color mapping. The latter was found by all participants to only contain 2 depth layers, which confirmed our earlier finding. Ten participants distinguished one more depth level for our mapping than for the hue. Only one participant identified 4 layers for the hue mapping and 3 for ours.

This informal user study matches our expectations and the assumption that our mapping leads to a better depth perception seems to hold. In addition, on structured models, like the 3D scene, our mapping is preferred and led to a better understanding of the 3D scene when compared to the previous hue mapping. The versatility of applications of this mapping is underlined via the point cloud example, in which more depth levels could be identified, and the blood vessels illustration, in which participants performed consistently better in identifying structures than with previous mappings.

7 Results

Fig. 12 shows a simple model, that illustrates the increased depth range and shows that a layering impression, as for red/magenta/blue is avoided. The same holds for complex models (Fig. 13), which lead to a more continuous appearance and a better depth separation.

Another application of our approach is cartography (Fig. 14), color charts are often used to encode depth in the 2D image. Although the different altitudes may be more difficult to discriminate using our instead of the hue mapping, a correspondence can be easily found between the two. Light red stands for yellow, white for green and light blue for cyan. On top of that, surface details are better perceived using lighter colors, as the luminance contrast is greater. Also, the smooth transition in our mapping avoids the layering effect of the hue mapping [Rheingans 1999]. Additionally, it encodes a stereoscopic effect when viewed using ChromaDepth®glasses. This backward compatibility is an interesting feature and makes stereo glasses optional but not mandatory. The same holds for scientific visualization (Fig. 15), where color fringing artifacts are reduced and a better depth discrimination is achieved.

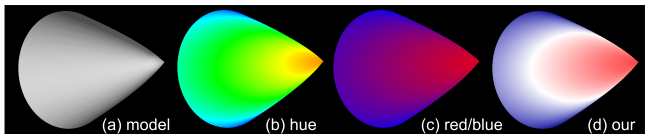


Figure 12: *(a) a 3D cone model with (b) linear depth to hue, (c) red to blue, (d) and our mapping.*

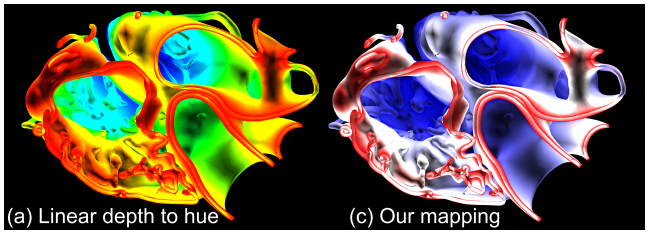


Figure 13: *Transversally-cut heart with a hue and our mapping.*

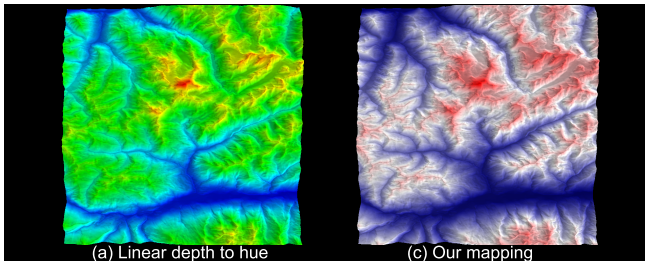


Figure 14: *Matterhorn: Our colors preserve more depth details.*

8 Conclusion

We presented a novel method to produce chromostereoscopic images. We introduce a novel color mapping to ensure a better depth perception which is compatible with shading computations. Using unsharp masking we can further enhance depth discontinuities and reduce cross talk between neighboring pixels. Our solution is fast

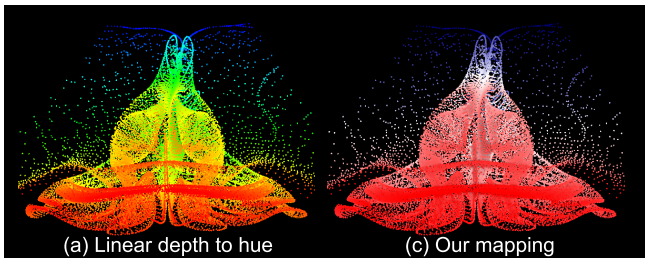


Figure 15: *Air flow around ellispoid.*

and easy to implement, which makes it a good candidate in various application contexts. Our solution is applicable to LCD screens, as well as printing technology and makes it possible to produce high quality output even on such devices that are theoretically incapable of producing all necessary wavelengths for a consistent depth perception. Our solution gives insights into the limitations of previous methods and might be key to an increased use and spread of chromostereoscopic images.

9 Acknowledgments

The work was partially funded by the EU FP7-323567 project Harvest4D, the Dutch KvK program, and the Intel VCI at Saarland University. We also like to thank K. Myszkowski for bringing chromostereoscopy to our attention.

References

ANSTIS, S. M., HOWARD, I. P., AND ROGERS, B. 1978. A craik-o'brien-cornsweet illusion for visual depth. *Vision Research 18*, 2, 213 – 217.

BAILEY, M., AND CLARK, D. 1998. Using chromadepth to obtain inexpensive single-image stereovision for scientific visualization. *Journal of Graphics Tools 3*, 3, 1–9.

BAILEY, R., GRIMM, C., AND DAVOLI, C. 2006. The effect of warm and cool object colors on depth ordering. In *Proc. App. Perception in Graphics and Visualization*, ACM, 161–161.

BROOKES, A., AND STEVENS, K. A. 1989. The analogy between stereo depth and brightness. *Perception 18*, 5, 601–14.

CHU, A., CHAN, W.-Y., GUO, J., PANG, W.-M., AND HENG, P.-A. 2008. Perception-aware depth cueing for illustrative vascular visualization. In *BMEI*, vol. 1, 341–346.

DIDYK, P., RITSCHEL, T., EISEMANN, E., MYSZKOWSKI, K., AND SEIDEL, H.-P. 2011. A perceptual model for disparity. In *ACM SIGGRAPH 2011 Papers*, 96:1–96:10.

DIDYK, P., RITSCHEL, T., EISEMANN, E., MYSZKOWSKI, K., SEIDEL, H.-P., AND MATUSIK, W. 2012. A luminance-contrast-aware disparity model and applications. *ACM Trans. Graph. 31*, 6 (Nov.), 184:1–184:10.

EINTHOVEN, W. 1885. Stereoscopie durch Farbendifferenz. *Albrecht von Graefes Archiv für Ophthalmologie 31*, 3, 211–238.

FAUBERT, J. 1994. Seeing depth in colour: More than just what meets the eyes. *Vision Research 34*, 9, 1165 – 1186.

GOOCH, A., GOOCH, B., SHIRLEY, P., AND COHEN, E. 1998. A non-photorealistic lighting model for automatic technical illustration. In *Proceedings Computer Graphics and Interactive Techniques*, ACM, 447–452.

HAMID, I. A. 2012. *Chromo-Stereoscopic visualisation for dynamic marine operations*. PhD thesis, University of Plymouth.

HECKMANN, T., AND SCHOR, C. 1989. Is edge information for stereoacuity spatially channeled? *Vision research 29*, 5, 593–607.

KEMELMACHER-SHLIZERMAN, I., SHECHTMAN, E., GARG, R., AND SEITZ, S. M. 2011. Exploring photobios. *ACM Trans. Graph. 30*, 4 (July), 61:1–61:10.

LEGGE, G. E., AND YUANCHAO, G. 1989. Stereopsis and contrast. *Vision Research 29*, 8, 989 – 1004.

LIVINGSTONE, M. 2002. *Vision and Art: The Biology of Seeing*. Harry N. Abrams.

LUFT, T., COLDITZ, C., AND DEUSSEN, O. 2006. Image enhancement by unsharp masking the depth buffer. *ACM Transactions on Graphics 25*, 3 (jul), 1206–1213.

OSKAM, T., HORNUNG, A., BOWLES, H., MITCHELL, K., AND GROSS, M. 2011. OSCAM - Optimized stereoscopic camera control for interactive 3D. *ACM Trans. Graph. 30*, 6.

PALMER, S. E. 1999. *Vision science : photons to phenomenology*. MIT Press, Cambridge, Mass.

PETRIE, G., TOUTIN, T., RAMMALI, H., AND LANCHON, C. 2001. Chromo-stereoscopy: 3D stereo with orthoimages & DEM data. *GeoInformatics 4*, 6, 8–11.

RHEINGANS, P. 1999. Task-based color scale design. In *In Proceedings Applied Image and Pattern Recognition. SPIE*, vol. 3905, 35–43.

ROPINSKI, T., STEINICKE, F., AND HINRICHS, K. 2006. Visually supporting depth perception in angiography imaging. In *Smart Graphics*, Springer, 93–104.

SICKING, J., TOEPKER, T., AND WOJTKIEWICZ, G. 1995. *The physics teacher 33*, 7, 446–448.

SIMONET, P., AND CAMPBELL, M. C. W. 1990. Effect of illuminance on the directions of chromostereopsis and transverse chromatic aberration observed with natural pupils. *Ophthalmic and Physiological Optics 10*, 3, 271–279.

STEENBLICK, R. A. 1987. The chromostereoscopic process: A novel single image stereoscopic process. In *True Three-Dim. Imaging Tech. and Display Tech.*, SPIE, vol. 0761, 27–34.

TOUTIN, T. 1997. Qualitative aspects of chromo-stereoscopy for depth perception. *Photogrammetric Engineering and Remote Sensing 63*, 2, 193–204.

UCKE, C. 1999. 3-D vision with chromadepth glasses. In *Proceedings of the ICPE/GIREP-Conference*, 23–28.

VERWICHTE, E., AND GALSGAARD, K. 1998. On the visualization of three-dimensional datasets. *Solar Physics 183*, 2, 445–448.

WALLISCH, B., MEYER, W., KANITSAR, A., AND GRLLER, E., 2002. Information highlighting by color dependent depth perception with chromo-stereoscopy.

WANG, C., AND SAWCHUK, A. A. 2008. Disparity manipulation for stereo images and video. vol. 6803, 68031E–68031E–12.

YE, M., BRADLEY, A., THIBOS, L. N., AND ZHANG, X. 1991. Interocular differences in transverse chromatic aberration determine chromostereopsis for small pupils. *Vision Research 31*, 10, 1787–1796.

Accelerated Mean Shift For Static And Streaming Environments

Daniel van der Ende*, Jean Marc Thiery†, and Elmar Eisemann‡
Delft University of Technology
Delft, The Netherlands
Email: *daniel.vanderende@gmail.com, †j.thiery@tudelft.nl, ‡e.eisemann@tudelft.nl

***Abstract*—Mean Shift is a well-known clustering algorithm that has attractive properties such as the ability to find non convex and local clusters even in high dimensional spaces, while remaining relatively insensitive to outliers. However, due to its poor computational performance, real-world applications are limited. In this article, we propose a novel acceleration strategy for the traditional Mean Shift algorithm, along with a two-layer strategy, resulting in a considerable performance increase, while maintaining high cluster quality. We also show how to to find clusters in a streaming environment with bounded memory, in which queries need to be answered at interactive rates, and for which no mean shift-based algorithm currently exists. Our online structure is updated at very minimal cost and as infrequently as possible, and we show how to detect the time at which an update needs to be triggered. Our technique is validated extensively in both static and streaming environments.**

Keywords–Data stream clustering; Mean Shift

I. Introduction

Although streams of data have been generated for a considerable amount of time, the analysis of these streams is a relatively young research field. Data streams present additional challenges for the field of data mining. Traditional data mining algorithms cannot be directly applied to data streams, due to the additional data stream constraints, which has led to considerable research into new methods of analyzing such high-speed data streams [1], [2].

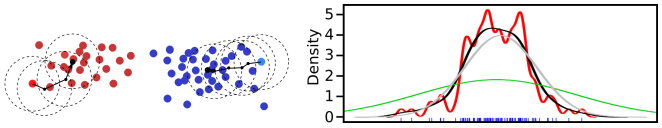
Mean Shift, initially proposed by Fukunaga et al. [3] and later generalized by Cheng et al. [4] and Comaniciu et al. [5], is a well-known clustering algorithm that has a number of attractive properties, such as its ability to find non-convex clusters. However, its performance has always been a concern, and it is because of this that, we believe, Mean Shift has never been applied in a streaming environment.

In this article, we present a modification of Mean Shift that can be used in both static and stream clustering environment. In the static environment, execution time of Mean Shift is reduced while a high level of cluster performance is maintained (Section III-A). Our main contribution concentrates on streaming environments, and we derive an efficient triggering mechanism, used to determine when a reclustering of the structure is necessary (Section III-B). We provide extensive experimental validation of our two contributions.

II. Background

A. Mean Shift

1) Overview: Mean Shift is a mode-seeking, density-based clustering technique, with as main parameter a *kernel bandwidth* h describing the scale at which clusters are expected. In this regard, Mean Shift can be seen as a natural multi-scale clustering strategy.



(a) Each point performs an iterative gradient ascent of the estimated density towards a local maximum.

(b) Estimated density (red, grey, black) using various bandwidths. Blue points are distributed according to the green density.

Figure 1. Mean Shift overview process

Considering an input data set $\mathcal{P} = \{\mathbf{p_i}\}$ in dimension d and a density kernel K, a Mean Shift clustering of $\mathcal{P}$ is obtained as follows: For every point $\mathbf{p_i}$, initialize $p_i^0 = \mathbf{p_i}$ and iteratively compute p_i^{k+1} from p_i^k by performing a gradient ascent of the density kernel. Upon convergence $p_i^\infty = \bar{p}_i$, where $\bar{p}_i$ is a local maximum of the density kernel. Points of $\mathcal{P}$, which converge towards the same local maximum are then clustered together. Figure 1(a) gives a schematic overview of this process.

It should be noted, that the underlying geometric structure of the clusters is of course dependent on the kernel that is used. In particular, changing the bandwidth of the kernel results in more or fewer local maxima of the resulting density kernel. Figure 1(b) illustrates this fact.

A variety of kernels has been used in the literature, the most common of which is the traditional Gaussian kernel (with *bandwidth* $h \in \mathbb{R}$):

$$K(x,p) = \pi^{-d/2} \exp(-||x-p||^2/h^2) \tag{1}$$

Note that this isotropic kernel is sometimes replaced by an anisotropic Gaussian kernel described by a symmetric positive matrix H (i. e., $\exp(-||x-p||^2/h^2)$ is replaced by $\exp(-(x-p)^{\mathrm{T}} \cdot H \cdot (x-p))$). However, if the anistropy of the kernel is uniform (i. e., $H(x) = H$ regardless of the location x), then these two approaches are completely equivalent.

Indeed, because H is symmetric definite positive, it can be decomposed as $H = U^{\mathrm{T}} \cdot \Sigma \cdot U$, U being a rotation matrix and Σ being a diagonal matrix with positive entries, which describe the different anisotropic scales of the kernel. Then, by noting $\sqrt{\Sigma}$ the diagonal matrix with squared scales ($\sqrt{\Sigma}^{\mathrm{T}} \cdot \sqrt{\Sigma} = \Sigma$), it is easy to verify that $(x-p)^{\mathrm{T}} \cdot H \cdot (x-p) = ||(x'-p')||^2$, with $y' := \sqrt{\Sigma} \cdot U \cdot y$ for every point y.

It is therefore entirely equivalent to use a uniform anisotropic kernel on the input data and use a uniform isotropic

kernel on data that has been globally transformed through the rigid transformation $y' := \sqrt{\Sigma} \cdot U \cdot y$. Note that traditionally, principal component analysis (PCA) [6] is a common strategy to first transform the input data before applying Mean Shift.

In our work, we will thus only focus on the case of isotropic Gaussian kernels.

2) Bandwidth estimation: The user may not always have an idea of what bandwidth to use, in the context of data which is difficult to explore, visualize and understand, such as high-dimensional data. There are a great number of bandwidth estimation techniques [7]–[10] providing results commonly accepted by the scientific community as *intrinsic* to the data. In this respect, Mean Shift can then be seen as a non-parametric clustering method. In our work, for datasets with non-provided bandwidth, we will use Silverman's rule of thumb [9].

3) Performance: Although Mean Shift has many attractive properties, such as its ability to find non-convex clusters and its multiscale nature, it also has some limitations and issues. The most important limitation is its performance. As Fashing et al. [11] have shown, Mean Shift is a quadratic bound maximization algorithm whose performance can be characterized as being $O(kN^2)$, where N is the number of points, and k is the number of iterations per point.

Many modifications to Mean Shift have been proposed [12]–[17]. Carreira-Perpiñán [12] identify two ways in which Mean Shift can be modified to improve performance: 1) Reduce the number of iterations, k, used for each point, 2) Reduce the cost per iteration. As Carreira-Perpiñán demonstrates, both of these techniques have their own merits and issues. Another class of Mean Shift modifications is that of data summarization, followed by traditional Mean Shift on a summary of the original data. Our algorithm falls into this category. This is an approach that other acceleration strategies have also applied [14], [16]. Further details on this will be given in Section III.

B. Data Stream Clustering

A number of authors have assessed the complexity of mining data streams [18], [19]. Barbara [19] focused on data stream clustering, listing a number of requirements: 1) Compactness of representation, 2) Fast, incremental processing of new data points, 3) Clear and fast identification of outliers. Due to the nature of streams, time is very limited. Because of this, data stream clustering algorithms need to be able to respond extremely quickly to the changes that occur over time in the dataset, often called concept drift. Moreover, because of the often huge datasets, memory is also constrained. Our approach has both attractive time and memory use characteristics, as will be discussed in Section III.

Many stream clustering algorithms use a two-phase approach. The approach centers on an online phase, which summarizes the data as it is streamed in, and an offline phase, which executes a given clustering algorithm on the summaries produced. The summaries are generally referred to as micro-clusters, and due to a number of attractive properties can be updated as time progresses and new data is streamed in (and old data is streamed out). CluStream [20] maintains q micro-clusters online, followed by a modified k-means algorithm that is executed when a clustering query arrives. DenStream [21] is similar, exchanging the k-means algorithm for DBSCAN, and distinguishing various quality levels of micro-clusters. Finally, D-Stream [22] uses a sparse grid approach. All these three algorithms have complex parameters. Moreover, when a clustering query arrives, a clustering is always executed.

The Massive Online Analysis (MOA) [23] framework offers a sandbox environment for easy comparison of several stream clustering algorithms. Among those, D-Stream [22] is the most related to our approach. Even though, D-Stream is not a Mean Shift algorithm. It is a density-based approach, and it partitions the space by computing the connected components of the set for which the local density is higher than a given threshold. Other parts of the space are considered outliers (Mean Shift offers a soft characterization of outliers, through the number of points in clusters and the value of the density at the clusters' center). It integrates a particular kind of density decaying mechanism, whereas our approach allows for various windowing strategies. Further, our main contribution is a new triggering mechanism, which detects events for when the clustering needs to be updated. Although not investigated, our triggering mechanism could be used for D-Stream as well. Note that the purpose of this paper is not to claim the superiority of Mean Shift over other existing clustering algorithms. Each algorithm has its advantages and drawbacks.

III. Method

A. Static Clustering

As discussed in Section II-A3, there are several ways in which previous work has improved Mean Shift. Our algorithm aims to reduce the number of input points for the Mean Shift. This is achieved by first discretizing the data space using a sparse d-dimensional regular grid, with a cell size of the order of the bandwidth (coarser discretizations lead to artifacts). For each grid cell C_i, the number of points n_i assigned to it is maintained, along with the sum of these points S_i. This enables the computation of an average position of the points within the cell, denoted $\overline{C_i} = S_i/n_i$. We then simply cluster the cells $\{C_i\}$ by applying the Mean Shift algorithm over them, using $K_{\overline{C}}(p, C_i) = n_i K(p, \overline{C_i})$ as the underlying kernel (see Algorithm 1). This is equivalent to computing the Mean Shift over all input points, after having set each point to the center of its cell. Although extremely simple, this strategy proved robust and efficient during our experiments. It also allows us to run Mean Shift over infinitely growing datasets with bounded memory, as long as the range of the data remains bounded, which is a required property in streaming environments.

Algorithm 1 Update clustering of cells $\{C_i\}$.

for all cell C_i **do**
  $\overline{C_i} = S_i/n_i$
end for
kdt = computeKdTree($\{\overline{C_i}\}$)
for all cell C_i **do**
  $\hat{c}_i = \overline{C_i}$
  for $it < ItMax$ **do**
    NN=kdt.nearestNeighbors($\hat{c}_i$)
    $\hat{c}_i = \sum_{k \in NN} n_k K(\hat{c}_i, \overline{C_k}) \overline{C_k} / \sum_{k \in NN} n_k K(\hat{c}_i, \overline{C_k})$
  end for
end for
cluster $\{C_i\}$ based on proximity of $\{\hat{c}_i\}$.

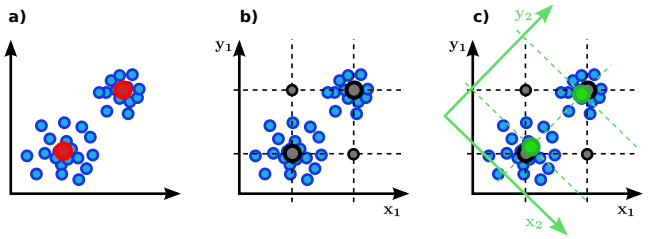


Figure 2. **a):** Two clusters in $2D$, with their centers in red. **b):** Clusters in each dimension (dot lines), whose product badly approximate the $2D$ clusters. **c):** Consensus over additional axes helps to identify the $2D$ clusters from the product of the $1D$ clusters of the projected data.

B. Stream Clustering

For stream clustering, the most common methodology is a two-phase approach, as discussed in Section II-B. A major disadvantage of this approach is that when a user query arrives, the offline clustering algorithm is executed over the data summaries, regardless of whether the dataset has changed significantly since the previous execution of the offline phase. This leads to unnecessary and expensive computations.

Our algorithm aims to avoid such needless clustering algorithm execution by accurately detecting when the data has changed sufficiently to warrant a new clustering. This is achieved by fast, effective analysis of the data currently being considered. It should be noted that this approach can be used, regardless of the type of window used. In this article, we have applied both landmark and sliding windows of various sizes.

First, when the stream clustering is initialized, a static clustering, as described in Section III-A is performed. This clustering will serve as our initial reference clustering. On each stream iteration, we evaluate whether the data distribution has changed sufficiently compared to this reference clustering to require a reclustering. If so, a clustering is executed and the result is considered the reference clustering for future iterations. By only executing the clustering algorithm, Mean Shift in our case, when necessary, a great amount of execution time is saved. Querying the cluster for a point is then done by finding the closest cell average and retrieving its cluster index (this requires an acceleration structure such as a Kd-Tree, which was already computed at the Mean Shift step).

We base our trigger mechanism on the monotonicity lemma defined by Agrawal et al. [24] as:

Lemma 3.1 (Agrawal): If a collection of points S is a cluster in a k-dimensional space, then S is also part of a cluster in any $(k-1)$-dimensional projections of this space.

Following this lemma, if any of the k-dimensional clusters change, a $(k-1)$-dimensional subcluster should also change. We make use of this point and set up a collection of low-dimensional *data observers* (in our case, $1D$), which we can update efficiently when adding or removing points from the structure, and which will trigger a reclustering of the structure when necessary. Our algorithm is parametrized by the chosen distribution of observers as well as by their *sensitivity*.

Each observer i is defined as an histogram $\mathbf{H_i}$ of the data projected onto an axis $\mathbf{a_i}$. Because high-dimensional data usually overlaps in separate dimensions (see Figure 2), we consider not only the canonical axes $\{\mathbf{e_k}\}$, but also randomly distributed axes in $\mathbb{R}^d$, and we will define the final decision for the reclustering as a consensus over the observers. Since we cannot make any assumption on the upcoming data (e. g., align data using PCA), we create a random set of pairs of indices $(k_1, k_2) \in [1, d]^2$ and define the additional axes as $(\mathbf{e_{k_1}} + \mathbf{e_{k_2}})/\sqrt{2}$ and $(\mathbf{e_{k_1}} - \mathbf{e_{k_2}})/\sqrt{2}$ (we thus *intricate* the canonical dimensions (k_1, k_2), see Figure 2**(c)**). All histograms are treated equally throughout.

When a clustering is performed, each histogram is saved as $\bar{\mathbf{H}}_\mathbf{i}$. On each subsequent stream iteration, data points are added to the grid (or removed from it if a time-dependent window is used), all histograms are updated, and we determine if the stream iteration has significantly altered the data distribution, in which case we need to update the clustering.

We define the measure between histograms $\bar{\mathbf{H}}_\mathbf{i}$ and $\mathbf{H_i}$ as their Jensen-Shannon divergence:

$$D_{JS}(\bar{\mathbf{H}}_\mathbf{i} \parallel \mathbf{H_i}) = \frac{1}{2} D_{KL}(\bar{\mathbf{H}}_\mathbf{i} \parallel M) + \frac{1}{2} D_{KL}(\mathbf{H_i} \parallel M) \quad (2)$$

where $M = \frac{1}{2}(\bar{\mathbf{H}}_\mathbf{i} + \mathbf{H_i})$, and $D_{KL}(P \parallel Q)$ is the Kullback-Leibler divergence between histograms P and Q:

$$D_{KL}(P \parallel Q) = \sum_k P(k) \ln \frac{P(k)}{Q(k)} \quad (3)$$

This measure is a distance, which is (symmetric and) always defined. Note that the direct use of the Kullback-Leibler divergence between $\bar{\mathbf{H}}_\mathbf{i}$ and $\mathbf{H_i}$ results in $+\infty$ in cases where points are removed from a cell, i. e., $Q(k) = 0$ in (3).

A histogram i *votes* for a reclustering if $D_{JS}(\bar{\mathbf{H}}_\mathbf{i} \parallel \mathbf{H_i}) > \epsilon$ (ϵ defines the *sensitivity*, which is our main input parameter).

A reclustering is then decided if the proportion of histogram voting for a reclustering is larger than a random variable, which we take between 0 and 1. This procedure is a standard Monte Carlo voting scheme, which will never (resp. always) trigger a reclustering if no (resp. all) histograms vote for it, and which will trigger a reclustering with probability defined by the consensus among the observers.

The procedure described above (for which pseudo-code is given in Algorithm 2) is easily maintainable in a streaming environment, as it only requires removal and addition of points to histograms, which can take place very quickly. Moreover, the discretization of the data space bounds the memory use in such a way that very large datasets and data streams can succinctly, but accurately be stored and used.

Algorithm 2 Add (remove) p during streaming.

Require: saved histograms $\{\bar{\mathbf{H}}_\mathbf{i}\}$, sensitivity ϵ
  grid $\leftarrow (\rightarrow)\ p$ ▷ update grid
  $n^{\text{vote}}_{\text{histo}} = 0$
  for all histogram $\mathbf{H_i}$ with axis $\mathbf{a_i}$ **do**
    $\mathbf{H_i} \leftarrow (\rightarrow)\ \mathbf{a_i}^{\mathsf{T}} \cdot p$ ▷ update $\mathbf{H_i}$
    $n^{\text{vote}}_{\text{histo}} += D_{JS}(\bar{\mathbf{H}}_\mathbf{i} \parallel \mathbf{H_i}) > \epsilon\ ?\ 1 : 0$ ▷ get vote of $\mathbf{H_i}$
  end for
  if $n^{\text{vote}}_{\text{histo}} > rand() * N^{\text{total}}_{\text{histo}}$ **then**
    for all histogram $\mathbf{H_i}$ **do** $\bar{\mathbf{H}}_\mathbf{i} = \mathbf{H_i}$ ▷ save $\mathbf{H_i}$ in $\bar{\mathbf{H}}_\mathbf{i}$
    end for
    require update of Mean Shift
  end if

IV. Empirical Results

A. Metrics

We have computed the following cluster validation metrics: Jaccard Index, Rand Index, Fowlkes-Mallows Index, Precision, Recall, F-Measure (see the work of Meila et al. [25]). These metrics are based on pair-wise comparison of points of a reference clustering A and a comparison clustering A'. All metrics assess whether A' correctly classified the relation between the points in each pair. We use these 6 metrics to quantify our results instead of simply picking one, because there is no real consensus on what is the correct metric between clusterings. Furthemore, the metrics we chose are common in the data clustering scientific community and will hopefully provide a real insight into the behaviour of our algorithms to the reader. A value of 0 indicates completely different clusterings whereas 1 indicates identical ones.

We implemented our method in Python, since it is cross-platform and integrates well with real-world systems.

For the static clustering experiments, we compared the traditional Mean Shift and our modified algorithms on the input data points (with the same input bandwidth).

For the stream clustering experiments, the metrics show the deviation between the clustering of the cell averages $\{\overline{C_i}\}$, when using the triggering mechanism or instead updating the clustering every time.

The reason for this choice (comparing clusterings of the averages instead of the original points) is simply practical: we could not run the computation of these metrics on huge datasets for every stream step in a reasonable amount of time. Fortunately, the depicted errors are over-conservative: the true errors are actually lower than the ones we show. Indeed, consider the case of false classification of a new point in a clustering of $100k$ points: it will have a minor impact on the metrics as it is an outlier in the data, however it will create a new grid average in our coarse summarization grid (e.g., summarized by 100 cells) and will therefore result in computed errors (based on the averages), which are much higher.

B. Static Clustering

In order to evaluate our algorithm's performance, a large number of datasets were used. For each dataset, we compare our method with the traditional Mean Shift. Figure 3 shows a comparison between our approach and traditional Mean Shift. Most metric values have a value of the order of 0.99. While there are some minor differences, these regard points which are at the boundaries between visible clusters or outliers. In general, higher errors occur for datasets presenting a high variability of the range over its various dimensions (see the remark on the equivalence between isotropic and anisotropic Mean Shift in Section II-A1).

We have conducted experiments in higher dimension. Although visually comparable, it is difficult to even assess the correctness of the Mean Shift clustering by projecting the data on a 2D space, due to overlap in the visualization. Table I summarizes our results on various datasets commonly found in the scientific literature.

While we experienced a reasonable gain in performance for small to reasonably big datasets, this is of small importance. Rather, we emphasize that our approach produces results which

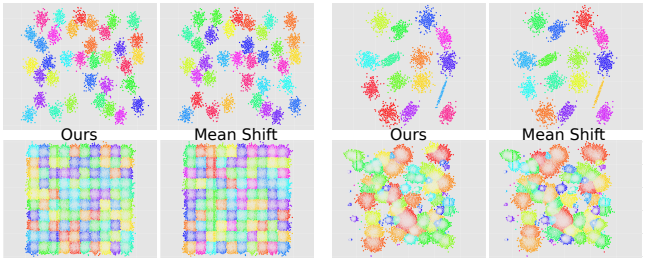


Figure 3. Comparison with Mean Shift on 2D data.

TABLE I. Summary of static clustering results. d: dimension. N: number of points. M1: Jaccard Index. M2: Fowlkes-Mallows Index. M3: Rand Index. M4: Precision. M5: Recall. M6: F-Measure

	d	N	M1	M2	M3	M4	M5	M6
A1	2	3000	0.95	0.97	0.99	0.97	0.97	0.97
A2	2	5250	0.96	0.98	0.99	0.98	0.98	0.98
A3	2	7500	0.96	0.98	0.99	0.98	0.98	0.98
S1	2	5000	0.99	0.99	0.99	0.99	0.99	0.99
S2	2	5000	0.95	0.98	0.99	0.98	0.97	0.98
S3	2	5000	0.87	0.93	0.99	0.95	0.91	0.93
S4	2	5000	0.85	0.92	0.99	0.94	0.90	0.92
Birch 1	2	100000	0.91	0.95	0.99	0.95	0.95	0.95
Birch 2	2	100000	0.64	0.78	0.95	0.76	0.99	0.78
Birch 3	2	100000	0.95	0.97	0.99	0.97	0.97	0.97
Dim 3	3	2026	0.99	0.99	0.99	0.99	0.99	0.99
Dim 4	4	2701	0.99	0.99	0.99	0.99	0.99	0.99
Dim 5	5	3376	0.99	0.99	0.99	0.99	0.99	0.99
D5	5	100000	0.99	0.99	0.99	0.99	0.99	0.99
Abalone	8	4177	0.99	0.99	0.99	0.99	0.99	0.99
D10	10	30000	0.99	0.99	0.99	0.99	0.99	0.99
D15	15	30000	0.99	0.99	0.99	0.99	0.99	0.99

are consistent with the traditional Mean Shift. This is the most important part of the validation, as it indicates that our approach can be used for Mean Shift clustering in a streaming environment, with potentially infinitely growing data. Note that, by construction, the error which is introduced by our approximation decreases with the size of the datasets on which it is used, while its efficiency obviously increases drastically.

C. Stream Clustering

For the stream clustering validation, we compare the results obtained when running our approach with the reclustering trigger enabled and disabled (i. e., reclustering on every stream iteration, regardless of lack of changes in the data distribution).

We show results on datasets of dimension 2 and 7, with a varying value of ϵ, with a fixed time window (with removal of old points) or not (adding points only), and with time-coherent or time-incoherent streaming of the data (i. e., whether the data is streamed in a structured way). Please note that "time-coherency" refers to the fact that points which are streamed in successively are roughly expected to belong to the same cluster. Table II summarizes the statistics of the conducted experiments, and the metrics plots for these test runs are presented in Figure 4. One interesting aspect with regards to our reclustering triggering is the value of ϵ which is used to threshold the Jensen-Shannon divergence value. For the CoverType dataset (dimension 7, $581k$ points), two test runs with identical configurations were performed, with $\epsilon = 0.001$ (Figure 4 (a)), and with $\epsilon = 0.003$ (Figure 4 (b)). The initial clustering was both times performed with $25k$ points, which is

TABLE II. Statistics of the experiments conducted in streaming environments. N: number of points. d: dimension. n: number of points for the initial clustering. δn: number of points added at each stream step. Window: number of points of the sliding window (if used). **TC**: time-coherency of the data stream.

	N (tot.)	d	ϵ	n (init.)	δn	Window	**TC**
a) Cov	$581k$	7	0.001	$25k$	$1k$	$25k$	NO
b) Cov	$581k$	7	0.003	$25k$	$1k$	$25k$	NO
c) Synth.1	$1M$	2	0.003	$50k$	$1k$	$50k$	NO
d) Synth.2	$1M$	2	0.003	$50k$	$1k$	$50k$	YES
e) Synth.3	$1M$	2	0.003	$50k$	$1k$	NO	YES

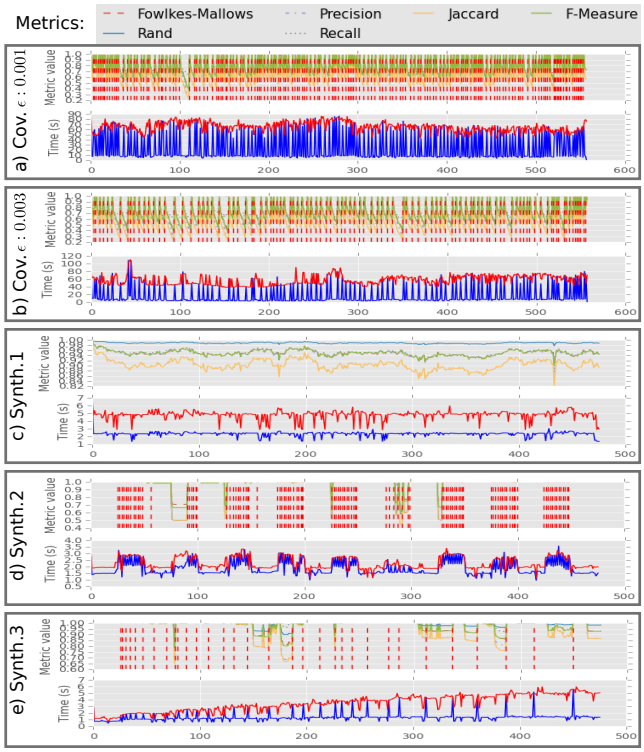


Figure 4. Comparison between our triggered clustering and constantly updated clustering on various datasets. For the timings, the red curve indicates the computation time per stream iteration when no triggering is used, and the blue curve indicates the one when our triggering mechanism is used. The vertical, dashed red lines indicate triggering events.

also the length of the sliding window that was used, and data points were streamed by sets of $1k$ points.

A lower value for ϵ should result in more frequent reclustering triggers, in an attempt to maintain a higher level of clustering quality. Although hard to see due to the high number of reclusterings, it is clear from these images that the lower ϵ affects the triggering mechanism. Reclusterings are more frequent and generally cluster quality is kept at a higher level. It should be noted that this dataset is also an example of a failure case of our algorithm, but also of the Mean Shift algorithm and any bandwidth-dependent algorithm. From the clustering results it is clear that the bandwidth estimate is completely incorrect. The bandwidth for this dataset was computed to be approximately 105.39. However, the CoverType dataset is not normally distributed, and the range of the data over the various dimensions varies from 1 to 109. Note for example that, on this dataset, a state-of-the-art Mean Shift grid approach implemented in Scikit-learn [26] provided results with metric values of 0.2 (with the same grid parameters).

For other experiments, the value of ϵ was set to $\epsilon = 0.003$.

Experiment Synth.1 (Figure 4 (c)) was done with a dataset of $1M$ points in dimension 2, with a sliding window of $25k$ points, with $1k$ points added at each stream and for time-incoherent streamed data. We observe that no reclustering is ever performed for this experiment. However, the metrics we obtain over time consistently remain over 0.7, which indicates that the initial clustering we had was good enough for the whole streaming session. Note that 0.7 is roughly the metrics values for which a reclustering was decided in previous experiments under similar conditions ($\epsilon = 0.003$), which indicates that the a-posteriori errors resulting from a given value of ϵ are consistent over the experiments.

Experiment Synth.2 (Figure 4 (d)), which was conducted under similar conditions as experiment Synth.1 with the sole difference of streaming time-coherent data, presents highly structured reclustering events. The reclustering events correspond to the appearance and disappearance of complete clusters, Our triggering mechanism visibly adapts in a non-trivial way to the structure of the underlying data.

Note that, for real-life datasets, the reality corresponds probably to a mix of these two behaviours (i. e., there are several levels of consistency in data, e. g., for visited websites during the day or in various places over the world, etc.). The strength of our approach is that we make no assumption on the structure of the data which is going to be streamed in, and that it adapts automatically to its underlying structure.

Finally, experiment Synth.3 was performed on a dataset of $1M$ points, without window (i. e., no points are removed). It is visible that the time for updating the structure grows almost linearly over time, while the frequence of the triggering events is actually inversely linear over time, which is the behaviour which is to be expected in order to provide timely-bounded analysis of growing data. Of course, there is a limit to this, and it is impossible to guarantee this behaviour for arbitrarily distributed data (over space and/or time).

V. Discussion

The experiments performed have shown that our algorithm produces accurate clusterings, at reduced cost, and only when necessary to maintain cluster quality. Moreover, our triggering mechanism allows Mean Shift to be applied in a streaming environment, which, to our knowledge, has not been achieved before. We now discuss possible extensions of our method.

First, it may be possible to apply a divide-and-conquer approach to the overall data space using the information contained in the histograms. In some cases, the clusters in the data space are clearly separated. It could then be useful to split the space, based on this information, into separate areas using cutting hyperplanes, and run our method on these distinct subspaces. This would allow for greater parallelism and avoid unnecessary work being done on clusters that do not change. Hinneburg et al. [27] developed a clustering algorithm based only on such cutting hyperplanes. Their technique for finding the optimal cutting hyperplanes could be applied to the sparse grid used in the approach discussed in this article.

Second, data sparseness can reduce the performance improvement over Mean Shift to some extent. In these extreme cases, if each point is placed in a grid cell of its own, running the Mean Shift on these cell averages $\overline{C_i}$ is effectively the same as running on the input data. This is also dependent on the bandwidth value used. Note however, that this effect appears mostly for "small" datasets. For very big datasets, which we target, it becomes improbable to keep a high degree of sparsity.

Third, as was discussed in Section II-A2, the bandwidth value h to a large extent determines the final clustering result. Thus, the quality of the results are also dependent on the quality of the bandwidth estimate or the value provided by the user. This sensitivity to the bandwidth parameter is an inherent problem for all approaches based on a kernel density estimate. An interesting avenue of research could be to maintain clusterings for various bandwidth values, and to use this information to derive a continuous clustering as the interpolation of the computed ones. Currently, if the bandwidth is reset by the user during streaming, our approach cannot update the clustering efficiently.

VI. Conclusion and Future Work

In this paper, we have presented an effective and efficient modification of the Mean Shift. The modification allows for faster execution when applied in a static context, and makes it possible to use Mean Shift in a streaming environment. The application of Mean Shift in a streaming environment is based on a two-phase approach, where a memory-efficient structure is maintained online, and Mean Shift is executed on this summary structure offline. Our triggering mechanism ensures that the expensive process of executing Mean Shift happens as infrequently as possible and only when necessary to ensure a high clustering quality. Only if the data distribution has changed strongly Mean Shift is executed again. Our approach is extensively validated in both the static and streaming environments, and shows good performance in both. We believe that our triggering mechanism might be usable for other stream algorithms. However, designing optimal mechanisms for specific stream algorithms as well as for specific error measures is an interesting lead for future work.

Acknowledgments

This work was partly funded by Mobile Professionals BV and EU FET Project Harvest4D.

References

[1] C. C. Aggarwal, Ed., Data Streams - Models and Algorithms, ser. Advances in Database Systems. Springer, 2007, vol. 31.

[2] J. A. Silva, E. R. Faria, R. C. Barros, E. R. Hruschka, A. C. de Carvalho, and J. Gama, "Data stream clustering: A survey," ACM Computing Surveys (CSUR), vol. 46, no. 1, 2013, p. 13.

[3] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," Information Theory, IEEE Transactions on, vol. 21, no. 1, 1975, pp. 32–40.

[4] Y. Cheng, "Mean shift, mode seeking, and clustering," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 17, no. 8, 1995, pp. 790–799.

[5] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 24, no. 5, 2002, pp. 603–619.

[6] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space," The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, vol. 2, no. 11, 1901, pp. 559–572.

[7] D. Comaniciu, V. Ramesh, and P. Meer, "The variable bandwidth mean shift and data-driven scale selection," in Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on, vol. 1. IEEE, 2001, pp. 438–445.

[8] D. Comaniciu, "An algorithm for data-driven bandwidth selection," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 25, no. 2, 2003, pp. 281–288.

[9] M. C. Jones, J. S. Marron, and S. J. Sheather, "A brief survey of bandwidth selection for density estimation," Journal of the American Statistical Association, vol. 91, no. 433, 1996, pp. 401–407.

[10] S. J. Sheather and M. C. Jones, "A reliable data-based bandwidth selection method for kernel density estimation," Journal of the Royal Statistical Society. Series B (Methodological), 1991, pp. 683–690.

[11] M. Fashing and C. Tomasi, "Mean shift is a bound optimization," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 3, 2005, pp. 471–474.

[12] M. A. Carreira-Perpinan, "Acceleration strategies for gaussian mean-shift image segmentation," in Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, vol. 1. IEEE, 2006, pp. 1160–1167.

[13] D. DeMenthon and R. Megret, Spatio-temporal segmentation of video by hierarchical mean shift analysis. Computer Vision Laboratory, Center for Automation Research, University of Maryland, 2002.

[14] D. Freedman and P. Kisilev, "Fast mean shift by compact density representation," in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009, pp. 1818–1825.

[15] B. Georgescu, I. Shimshoni, and P. Meer, "Mean shift based clustering in high dimensions: A texture classification example," in Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on. IEEE, 2003, pp. 456–463.

[16] H. Guo, P. Guo, and H. Lu, "A fast mean shift procedure with new iteration strategy and re-sampling," in Systems, Man and Cybernetics, 2006. SMC'06. IEEE International Conference on, vol. 3. IEEE, 2006, pp. 2385–2389.

[17] X.-T. Yuan, B.-G. Hu, and R. He, "Agglomerative mean-shift clustering," Knowledge and Data Engineering, IEEE Transactions on, vol. 24, no. 2, 2012, pp. 209–219.

[18] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, "Models and issues in data stream systems," in Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. ACM, 2002, pp. 1–16.

[19] D. Barbará, "Requirements for clustering data streams," ACM SIGKDD Explorations Newsletter, vol. 3, no. 2, 2002, pp. 23–27.

[20] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A framework for clustering evolving data streams," in Proceedings of the 29th international conference on Very large data bases-Volume 29. VLDB Endowment, 2003, pp. 81–92.

[21] F. Cao, M. Ester, W. Qian, and A. Zhou, "Density-based clustering over an evolving data stream with noise." in SDM, vol. 6. SIAM, 2006, pp. 326–337.

[22] Y. Chen and L. Tu, "Density-based clustering for real-time stream data," in Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2007, pp. 133–142.

[23] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, "Moa: Massive online analysis," The Journal of Machine Learning Research, vol. 11, 2010, pp. 1601–1604.

[24] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, Automatic subspace clustering of high dimensional data for data mining applications. ACM, 1998, vol. 27.

[25] M. Meilă, "Comparing clusteringsan information based distance," Journal of Multivariate Analysis, vol. 98, no. 5, 2007, pp. 873–895.

[26] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, 2011, pp. 2825–2830.

[27] A. Hinneburg and D. A. Keim, "Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering," in Proceedings of the 25th International Conference on Very Large Data Bases, ser. VLDB '99, 1999.