Space-Time Volume Visualization of Gaze and Stimulus

Valentin Bruder
University of Stuttgart
Stuttgart, Germany
valentin.bruder@vis.uni-stuttgart.de

Kuno Kurzhals
ETH Zurich
Zurich, Switzerland
kunok@ethz.ch

Steffen Frey
University of Stuttgart
Stuttgart, Germany
steffen.frey@vis.uni-stuttgart.de

Daniel Weiskopf
University of Stuttgart
Stuttgart, Germany
daniel.weiskopf@vis.uni-stuttgart.de

Thomas Ertl
University of Stuttgart
Stuttgart, Germany
thomas.ertl@vis.uni-stuttgart.de

ABSTRACT

We present a method for the spatio-temporal analysis of gaze data from multiple participants in the context of a video stimulus. For such data, an overview of the recorded patterns is important to identify common viewing behavior (such as attentional synchrony) and outliers. We adopt the approach of space-time cube visualization, which extends the spatial dimensions of the stimulus by time as the third dimension. Previous work mainly handled eye-tracking data in the space-time cube as point cloud, providing no information about the stimulus context. This paper presents a novel visualization technique that combines gaze data, a dynamic stimulus, and optical flow with volume rendering to derive an overview of the data with contextual information. With specifically designed transfer functions, we emphasize different data aspects, making the visualization suitable for explorative analysis and for illustrative support of statistical findings alike.

CCS CONCEPTS

• **Human-centered computing → Visualization techniques**;

KEYWORDS

Eye tracking, volume visualization, space-time cube

ACM Reference Format:
Valentin Bruder, Kuno Kurzhals, Steffen Frey, Daniel Weiskopf, and Thomas Ertl. 2019. Space-Time Volume Visualization of Gaze and Stimulus. In *Proceedings of ETRA conference (ETRA'19).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Gaze data recorded from multiple participants watching dynamic stimuli, such as videos, poses a challenge for eyetracking researchers. Complex spatio-temporal patterns that might appear in the data are hard to capture with statistical methods alone and often require visual support for (1) explorative data analysis, (2) displaying statistical results, and (3) the illustration of the results.

Established visualization techniques such as gaze plots and heat maps are limited for these purposes because they require animation to represent changing gaze patterns over time. In contrast, we aim for a static overview of gaze data from videos that conveys important contextual information and allows for efficient navigation in the data. To achieve this goal, we combine a space-time cube (STC) representation of the data with techniques known from volume rendering as it is used, for example, in medical imaging or simulation data analysis. Applying multiple transfer functions, we can combine data aspects for filtering and emphasizing important regions and time spans in the data.

Our contribution is the introduction of a visualization approach that combines multiple space-time volumes (video, optical flow, gaze) into a spatio-temporal overview that conveys gaze patterns as well as information on what caused these patterns. To this end, we propose specifically designed transfer functions that can reveal different aspects in the data. We demonstrate the applicability of our approach on various videos with gaze data from multiple participants, using our GPU-accelerated implementation of the system.

2 RELATED WORK

Related work for this paper is divided in a discussion of how space-time cubes are applied in eye tracking and more generally how other volume rendering techniques relate to this work. Eye tracking data visualizations can be separated in techniques that represent the point-based data directly, and the ones that include semantic information from Areas of Interest (AOIs) [2]. The presented technique is a point-based approach that visualizes raw gaze data. However, we aim to bridge the gap between point-based and AOI-based techniques by including stimulus content in the visualization. Hence, the semantic interpretation *what* a participant investigated *when*, is possible without pre-annotated AOIs.

Space-Time Cubes. Blascheck et al. [2] list the STC as a visualization for gaze data that provides spatio-temporal information in a static overview. Different approaches have been presented in recent years [21, 24, 26], all of them either showing gaze patterns or AOI-related information in an abstracted overview that still required additional video skimming to identify *what* happened in the stimulus when a visual pattern occurred. The STC originated from the analysis of geo-related events [15] and is often applied for the analysis of trajectories [14], similar to scanpaths in eye tracking. For 2D temporal field data, Frey et al. [13] proposed a visualization technique to help detect temporal patterns such as periodic changes or constant regions in the data. Other approaches for video-based graphics and video visualization [3] work with the video data directly to represent what is happening: Chen et al. [7] extract and visualize motion patterns from surveillance videos in

ETRA'19, June 2019, Denver, Colorado
2019. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

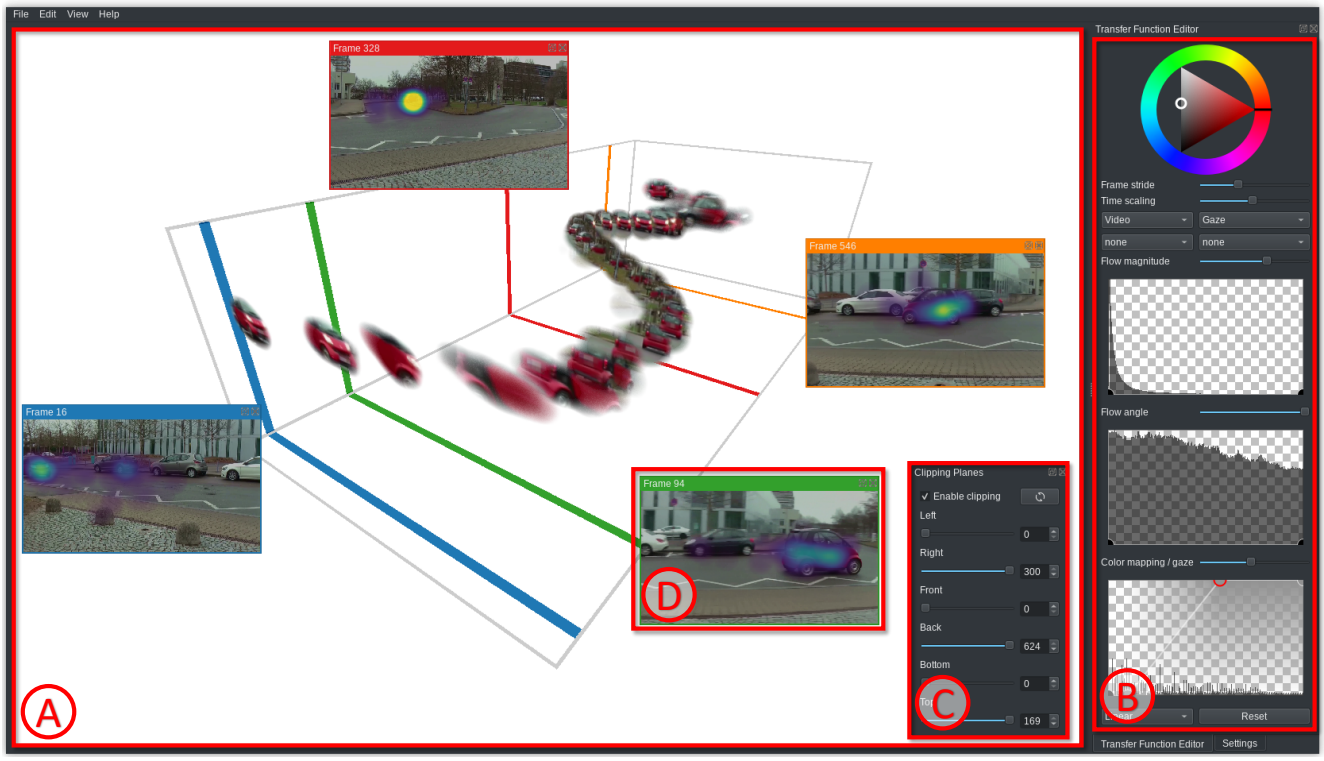


Figure 1: Our application combines: (A) the main view with the space-time cube visualization, (B) a transfer function editor for filtering and coloring that also shows data histograms, (C) an editor to adjust clipping planes, and (D) The visualization of selected frames as annotation that also shows the gaze heatmap.

an STC based on optical flow. A similar approach by Romero et al. [29], relies on heat map volumes to depict motion patterns in long-term recordings. These approaches do not depict the original video content in the resulting volume. Nguyen et al. [27] also use motion information to remove non-moving pixels and apply a slice-based representation of the parts of the video in motion. Hilpoltsteiner [16] introduced the idea of using gaze positions to reduce the depicted content in image slices inside the STC. However, the concept does not include the application of combined transfer functions for multiple data properties. To this point, a holistic approach supporting the visualization of the different data sources together is missing. With our approach, we bridge this gap between the analysis of gaze distributions and their interpretation by inspecting the video.

Volume Rendering. Traditional application domains of volume rendering techniques [8, 17] include medical imaging (analysis of computed tomography scans or magnetic resonance imaging), visual arts as in computer-generated imagery, and visualization of scientific simulation data (e.g., from astrophysics or material sciences). Since the advent of flexibly programmable graphics hardware, GPU raycasting has established itself as the standard for performing realtime direct volume rendering in workstation environments [9, 28, 31]. Volume raycasting has several advantages compared to rasterization, especially in the context of this work. It offers high quality and versatility in that all samples taken along the rays can be adjusted as desired and no ordering has to be applied to graphical elements. As it is often computationally expensive, enhancing the performance via different approaches has emerged into a significant field of research on its own [1, 5].

The visualization of temporal data is also an active field of research in volume visualization. Similar to the STC approach proposed in this paper that needs to be able to present multiple 3D fields, in these areas different forms of data reduction need to be employed to reduce overdraw and visual clutter. Woodring et al. [32] combine data from multiple fields into single values by using extended interaction modalities from standard volume rendering. This approach is conceptually similar from the perspective that we also rely on modified variants of standard volume exploration techniques. Another reduction approach is based on quantifying distances between time points and on this basis selecting single time steps to present to the user [11, 12], or extracting certain features like space-time discontinuities [10].Beyond classical applications of volume visualization for spatial field data, the advantages of volume rendering have also been exploited for visual analysis of large dynamic graph data [6].

3 TECHNIQUE

To explain our approach for combined analysis of gaze and stimulus data, we first discuss the visual design, followed by three core aspects that are depicted in Figure 2: (1) data pre-processing, (2) volume rendering, and (3) interactive data exploration.

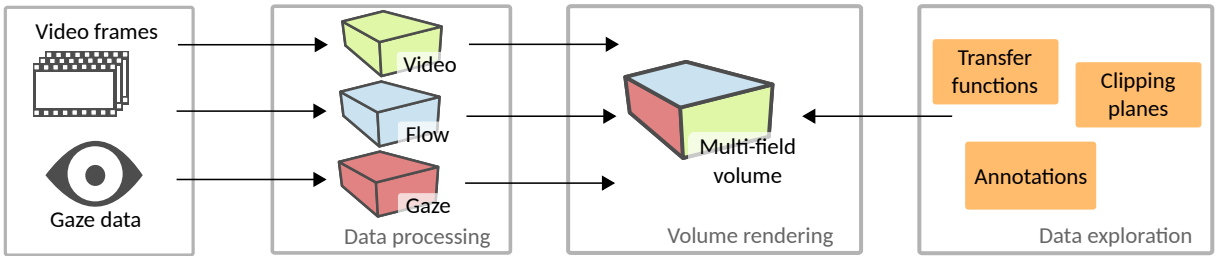


Figure 2: The stimulus video and the recorded gaze data from multiple participants is processed to derive three volumes containing spatio-temporal information of the data. Rendering is performed with a multi-field approach, combining the volumes. Interactive data exploration is supported via transfer functions, clipping planes, and annotations for the temporal dimension.

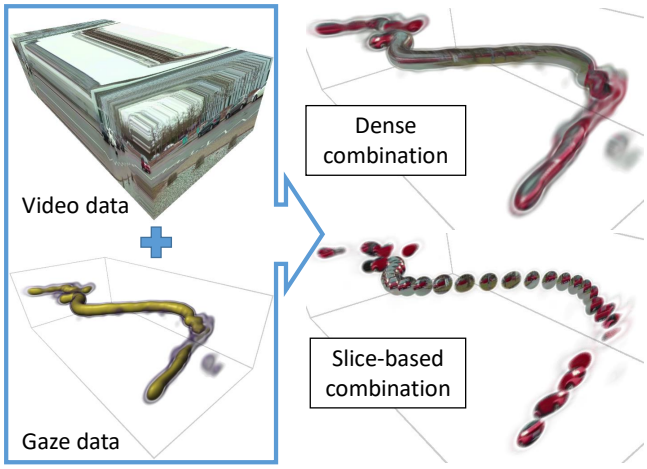


Figure 3: Example renderings of our data input (gaze density and video frames) as volumes and a combination thereof.

3.1 Visual Design

Figure 1 shows an overview of our system. It features the main view with the STC visualization (A), controls for filtering and highlighting parts of the data (B, C), and the possibility to annotate important frames (D). The two data sources for our visualization are video frames and gaze positions from multiple participants. Each data source can be investigated separately in the STC as depicted in Figure 3. The raw video volume provides an overview of motion at frame borders. This corresponds to slit-scan visualizations that are used to summarize a video [18]. However, important content inside the volume is occluded. Figure 3 also shows the aggregated gaze data rendered as an STC. While this visualization gives a good overview on the gaze distribution, there is no direct link to the video content. To make this connection visible, a combination of the two data sources is necessary. As shown in Figure 3 such a combination can be represented as a dense volume or slices, to reduce occlusion and reveal more details.

To support effective analysis of the combined data sources, preprocessing is required to transfer the data into a unified multi-field volume (Section 3.2). To render the volume interactively, we use GPU-accelerated raycasting (Section 3.3). For appropriate representation of important gaze patterns, we support interactive data exploration. Thereby, a key aspect is the manipulation of transfer functions to change the visualization based on different aspects in the data (Section 3.4).

3.2 Data Pre-Processing

To yield real-time rendering performance for interactive analysis, we convert the data into dense volumes in a pre-processing step.

Video Volumes. A video volume contains all visual information of the stimulus, and interaction methods from volume visualization can be employed. (1) The spatial plane depicts the video frames. Clipping the volume along the time axis emulates a video replay. (2) The side planes of the volume represent slit-scans of the video. Adjusting the clipping for these planes provides the corresponding slit-scan that can show important regions and events of the video.

The investigated stimuli consist of n frames, often in typical multimedia resolutions, e.g., 1920×1080 pixels. Including the temporal dimension in the data significantly increases the amount of memory necessary to hold such data. Hence, a reduction of the visualized data becomes necessary. For the representation of the data as a volume, the temporal resolution is more important because it allows depicting longer sequences, while the spatial resolution can be reduced without drastically changing the overview of the data set. However, with a width of only 100 pixels, the image content becomes blurry and details can be missed. Hence, we found that processing the videos for all frames with a width between 200–400 pixels and a height adjusted with respect to the aspect ratio of the video provides smaller volumes with enough details to interpret the stimulus. Furthermore, the ratio between the spatial and the temporal resolution increases with lower spatial resolutions, leading to elongated, less compact STCs, which might require additional scaling of the temporal axis for a better overview. The data is stored as RGBA unsigned characters in raw data files. This format is compatible with most applications for volume rendering. Note that the alpha channel in this format could be used to also store the gaze volume at the cost of precision.

Optical Flow. The optical flow for a video sequence describes how individual pixels move between two consecutive frames. We apply a variational method [4] that provides a dense vector field of absolute displacement for image pairs in the video sequence. For a more intuitive interpretation, the values are converted to angle and magnitude of the vectors. For filtering of motion regions,

Table 1: Three volumes are derived from the video and gaze data. They are combined in a subsequent rendering step.

Volume	Format	Channels	Content
Video	UCHAR	4	RGBA values of the video frames
Optical Flow	FLOAT	2	Angle & magnitude of the displacement vector field
Gaze	FLOAT	1	Gaze density based on kernel density estimation

pixel-precise accuracy of the flow is less important, hence, a trade-off between flow precision and computational performance can be made by reducing the number of iterations for the applied approach. We store the angle and magnitude as single precision floats in two channels. The normalization factor of the magnitude can be dynamically adjusted during runtime according to the needs of the analyst and specifics of the data.

Gaze Volumes. Heat maps are a common visualization to represent aggregated gaze data. The aggregation is calculated over time and/or for multiple participants. For dynamic stimuli, it is necessary to provide a dynamic heat map that conveys the changes of gaze patterns over time. To achieve this, we apply a sliding window approach that respects temporal coherence by summarizing gaze points from the current frame and $m \in \mathbb{N}$ previous frames. For the heat map calculation, we apply an Epanechnikov kernel [30] for an efficient approximation of a Gaussian kernel. The kernel covers 10% of the frame height, which roughly corresponds to the foveal area that was covered at a distance of 65 cm showing the videos with a resolution of 1920×1080 pixels. As with the optical flow, the applied techniques are interchangeable according to the requirements for precision and performance. The data is stored as single-precision floats without normalization. Again, the normalization of the data can be adjusted in the rendering process, allowing the analyst to change the heat map dynamically, depending on the current research question. Table 1 summarizes the data volumes and how they are stored for volume rendering.

3.3 Volume Rendering

We apply multi-field volume rendering to depict the three spatio-temporal volumes. To enable interactive exploration of the data, even when rendering large sequences with several thousand frames, we accelerate the compute-heavy calculations by using parallel processing on GPUs. For this, we use the OpenCL framework that allows for cross-platform execution and device portability. We also take advantage of texture units integrated in GPUs for their interpolation capabilities. Therefore, we process the video volume, optical flow, and gaze data as a 3D texture each.

Ray Casting. We implement direct volume rendering by using front-to-back raycasting. Therefore, we shoot rays through each pixel of the image plane toward the volume, sampling the volume elements (voxels) along the rays in equidistant intervals. For correct compositing and better performance, we sample all values (video, flow, gaze) at the same time. We use early ray termination and empty space skipping to accelerate the computations.

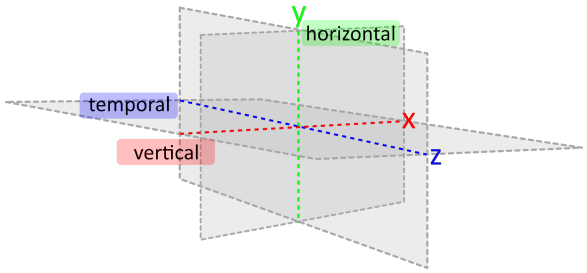


Figure 4: Clipping planes for exploration of volume slices.

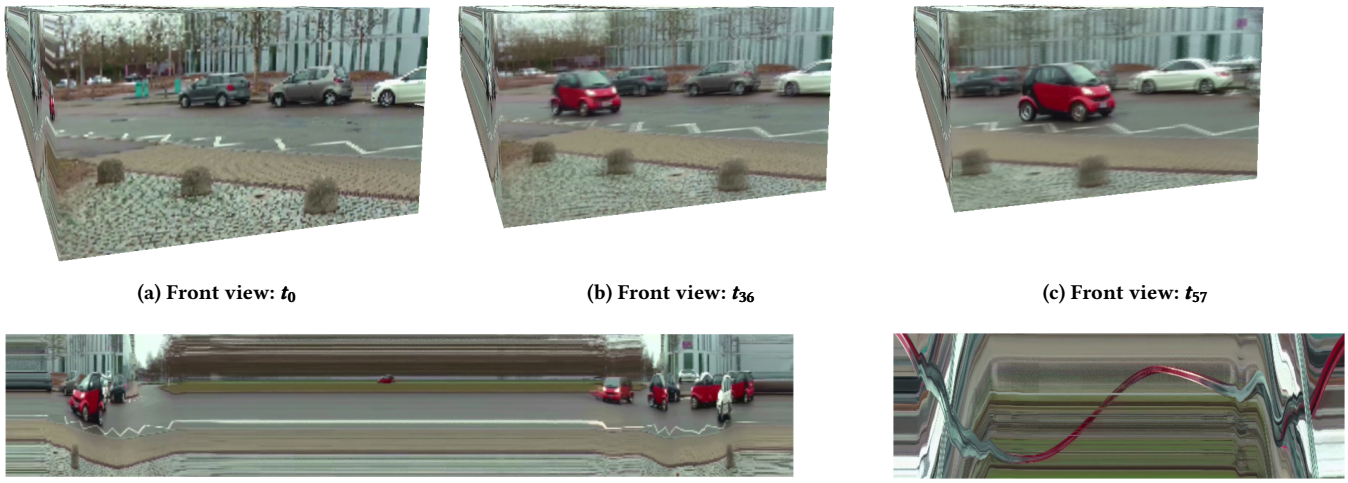
Multi-Field Rendering. In case of the video data, the sampled RGB colors are directly used to determine the color of the pixel. Density and optical flow data (angle and magnitude) are interpreted as three distinct scalar fields. The samples from these fields are evaluated using transfer functions. Being a fundamental concept in volume rendering, transfer functions map scalar values (which are sampled from the scalar field during raycasting) to color and/or opacity values. Thereby, mapped color and opacity are composed into the final pixel value. We support three transfer functions, one for color and opacity values and two for transparency only. If multiple transfer functions are employed at the same time, the opacity values are composited into one final opacity value. The sequence in which the functions are applied to the data (and if at all) can be dynamically adjusted by the analyst.

Slice-Based Video Context. The dense nature of the data makes it hard to make out the content of single frames, especially if little transparency is used in the mappings. To provide a meaningful visual representation nevertheless, we support rendering only selected frames in a regular interval, which can be dynamically adjusted. In Figure 1 for instance, a stride of 15 is used to show only data of 42 frames out of more than 620, which makes it possible to see most parts of the content for the rendered frames.

3.4 Interactive Data Exploration

The presented multi-field approach allows us to filter specific parts of the data and to emphasize regions in the STC that are relevant for the research question at hand. For example, one could only be interested to see when participants looked at moving objects. Combining the three presented data properties, such a query can be modeled by combining transfer functions. Furthermore, the application of clipping planes and timeline annotations provides means to explore the data and create supportive illustrations. Basic interactions along the temporal domain are supported such as scaling the data along the time-axis and showing only every n-th frame.

Clipping Planes. Clipping of the volume can be performed individually for each dimension (Figure 4). If clipped along the z-axis, the volume depicts how the video content changes over time, i.e., this corresponds to a playback of the video (Figures 5a–5c). If clipped along the x- or y-axis, the volume depicts individual slit-scans [18, 23] that summarize all motion over time at the clipping border, which acts as a scanline. This helps identify when an object appeared in the video (Figure 5d) or how it moved (Figure 5e).



(a) Front view: t_0 (b) Front view: t_{36} (c) Front view: t_{57}

(d) Side view: vertical plane clipping (e) Top view: horizontal plane clipping

Figure 5: Volume Clipping along the three dimensions. (a)–(c) Temporal clipping emulates a playback of the video. (d) Clipping the volume with a vertical plane results in slit-scans that depict objects whenever they moved through the scene. (e) Similarly, clipping with a horizontal plane reveals motion patterns.

Transfer Functions. Transfer functions determine the visual mapping of voxels to values such as color and opacity. A common approach to transfer function design is to select value ranges and their corresponding opacity based on a 2D histogram. For example, values that correspond to the hue of the sky in a video volume could be set to full transparency to remove one important area that often occludes important details in the volume. This corresponds to chroma keying techniques known from visual effects in video production. More general, the flow contains information that helps filter the data further. The histogram of the magnitude is helpful to remove areas with no motion, as with the example for the hue, regions such as the static sky can be masked out this way. Furthermore, general camera panning motion can be removed by appropriate filter settings (Figure 6b). Filtering the gaze data by its density allows us to highlight hotspots of attentional synchrony where the gaze density is high (Figure 6a). Regions with lower density values can also be emphasized, which is usually the case if multiple regions attracted attention, or if the gaze data is dispersed.

Annotations. Filtering the volume with appropriate transfer functions provides a better overview of the data set because of the local stimulus context, compared to visualizations without stimulus information. However, for the illustration of results, the global context, i.e., the whole video frame and the corresponding heat map are beneficial. Hence, we adapt the idea of annotations for narratives of historical events in the STC [19]. Kraak and Kveladze annotate events in a geo-spatial context by pictorial labels to summarize important events. We support such an interactive labeling of individual time steps by individually adjustable video frames. Therefore, the analyst can simply click on a frame to select in the space-time-cube directly. This invokes the generation of a hovering window containing the frame with a colored border that matches an also generated marker on the sides of the volumetric view that highlights annotated frames (Figure 1D). In future work, we want to add automated placement of the annotations and improve the visualization of the connections to the space-time view.

Table 2: Example videos with gaze data used in this paper.

Video	Duration	Depicted in figure
Car Pursuit	0:25 min	1, 3, 5, 6, 10
Kite	1:37 min	7
Thimblerig	0:30 min	8
UNO game	2:01 min	9

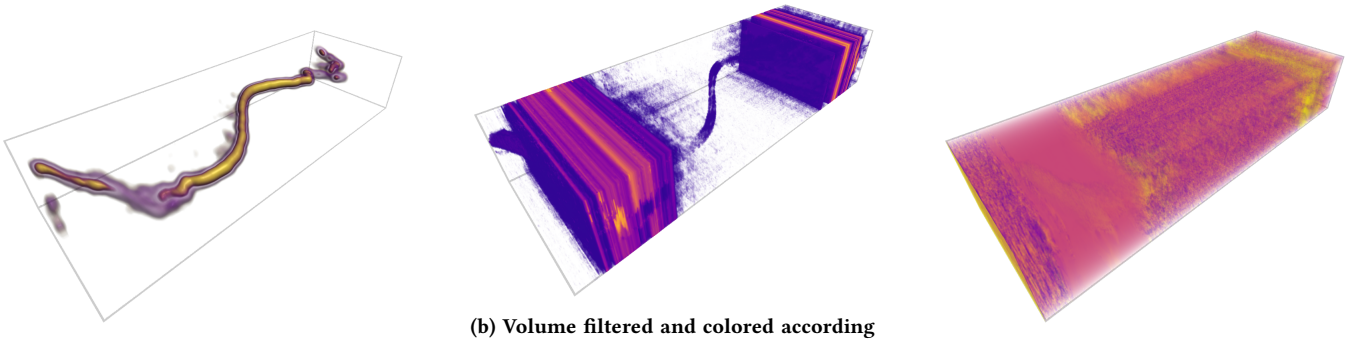
4 EXAMPLES

We apply our technique to different videos from a publicly available data set [20]. The videos with gaze data and the respective figures they are depicted in, are summarized in Table 2. All data was recorded in a user study with 25 participants using a Tobii Pro T60 XL with a stimulus resolution of 1920 × 1080 pixels. The *Car Pursuit* video has been shown in the previous sections to illustrate concepts of our technique. It depicts a red car driving from the left side of the screen to the right and back. The video contains two horizontal panning motions at the beginning and at the end to adjust the field of view. Participants were tasked to follow the car with their eyes, leading mainly to smooth pursuits and attentional synchrony [25] on the car.

4.1 Kite

Figure 7 depicts an example of our technique applied to a video where a person on a meadow steers a yellow kite. During the video sequence, the kite repeatedly leaves the recorded field of view. The participants were asked to follow the path of the flying kite.

Filtering out low gaze density from the video with the transfer function directly reveals patterns and outliers, thereby giving a

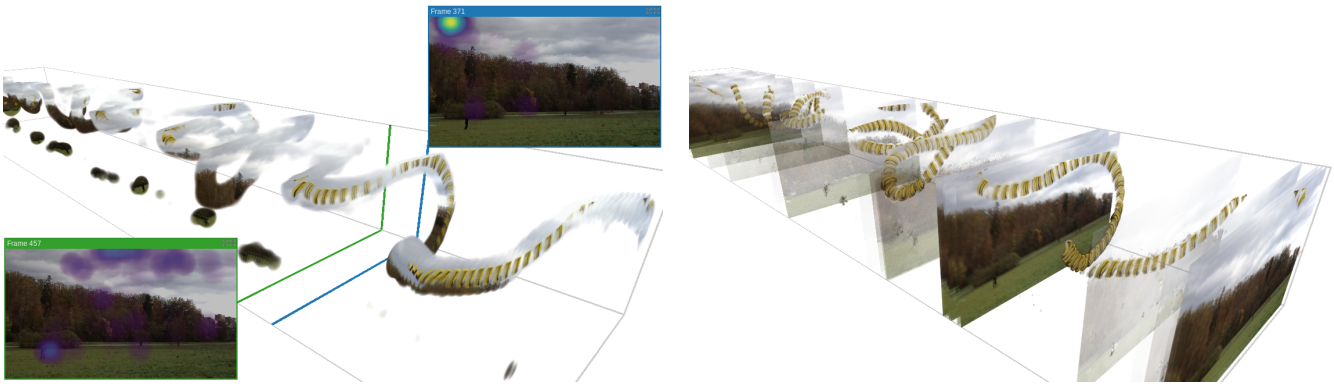


(a) Volume filtered and colored according to the gaze density

(b) Volume filtered and colored according to the transfer function for flow vector magnitude

(c) Volume filtered and colored according to the angle of the optical flow vectors

Figure 6: Different transfer functions for optical flow and gaze density applied to the same data set. Two camera pans and the moving car are clearly visible in the flow data, the gaze data shows that participants follow the car's movement accurately.



(a) Video data with low gaze density filtered out. Selected frames before and after the kite leaves the field of view.

(b) Video data with slow motion filtered out. Small camera pans are visible (not filtered out).

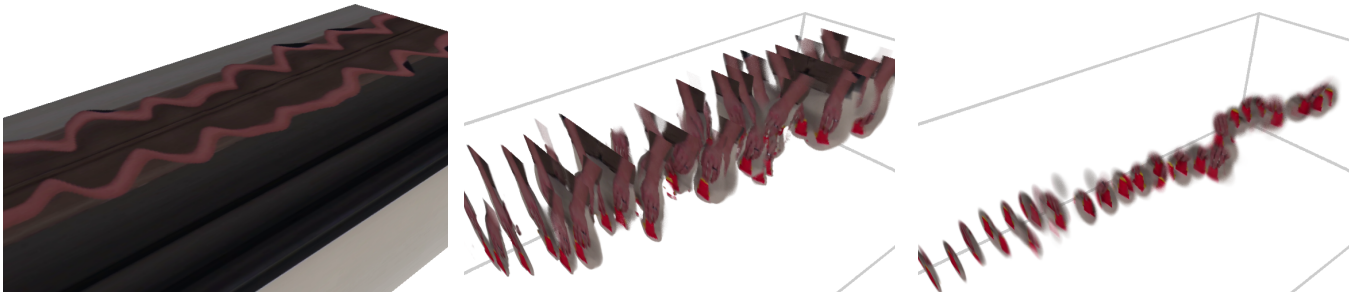
Figure 7: Rendering showing a video of a person steering a yellow kite. Filtering based on gaze as well as flow are applied to highlight different aspects.

good overview (Figure 7a). Mostly, participants smoothly follow the path of the kite if it is visible. However, if it leaves the field of view, some participants try to estimate the path outside the view and predict the spot where the kite reenters into the video. Other participants focus on the person steering the kite on the meadow. Figure 7b shows a rendering of the video while using our technique to filter based on the magnitude of the flow vectors (i.e., filtering out low magnitude). This rendering also reveals the path of the kite but also highlights the sections where the kite leaves the field of view. Slight camera pans in the data appear as fully visible frame slices but could be filtered out using another pre-processing step. This rendering also shows that the person on the ground is hardly moving (only slight indications) making it hard to identify the person as a potential AOI solely based on motion information. This example shows that using both, flow and gaze data for filtering has a clear advantage to visually identify AOIs.

4.2 Thimblerig

In our second example, we apply our technique to a video showing a hat game (thimblerig). The participants were asked to follow one of three hats that hides a marble underneath while they are being shuffled. Figure 8 shows renderings of the video data, which contains 749 frames, with different configurations. Looking only at the video data without any filtering applied (Figure 8a), the shuffling pattern is roughly visible.

Filtering out regions with low flow magnitude yields a concise overview of the shuffling patterns. The patterns can be enhanced using our slice-based video context view that regularly skips several frames as described in Section 3.3 (Figure 8b). Alternatively, we can also filter out regions with low gaze density (Figure 8c). This reveals that most participants followed a single hat—the one hiding the marble. Investigating the frames before the hat with the marble is lifted reveals that most participants were successful in following the hidden object. Comparing the two renderings with gaze respective motion filtered out, it becomes clearly visible that the gaze is directed by motion.



(a) Video with shuffling pattern (b) Frames with low motion filtered out (c) Frames with low gaze density filtered out

Figure 8: Video showing a hat game without and with filtering applied for different properties. Participants are tasked to follow the hat hiding a marble.

4.3 UNO Card Game

In this example, we apply our technique to a video of two people playing the UNO card game, until the right player wins. During each player's turn, participants were asked to focus on playable cards in the hands of the players.

Filtering the video data by omitting regions with low gaze provides a good overview of the major patterns (cf. Figure 9a). For instance, it is clearly visible when the majority of the participants looked either at the cards of the right or the left player. Typically, attention on one of the players is followed by gaze on the discard pile, then on the opposing player. This seems to be the main pattern in the gaze data. Participants follow the card, a player puts on the discard pile, and then looks at the other player to anticipate the next move. Visualizing the gaze data in the STC also reveals four time spans where the participants focus on the draw stack. Further investigation shows that all of these are related to events where players have to draw new cards. This can be visualized by filtering out regions with little motion (Figure 9b). Using this filtering, one can determine which player had to draw cards and even how many.

A close comparison of flow magnitude filtering that shows only parts with large motion and filtering with respect to high gaze density reveals aspects of interest. For instance, after the player on the right hand side plays the red card 1, the opponent cannot play a valid card and is forced to draw from the stack. Figure 9a shows that participants who watch the video react to this event differently. Some anticipate the draw as the next action, which is visible in the gaze visualization in that it shows attention on the draw stack before the motion of the hand begins, as can be seen in the flow visualization. However, most of the participants seem to follow the motion of the hand to the draw stack. This indicates that a few participants followed the game attentively and are able to anticipate the next move correctly before it happens. The attention of the other participants who follow the hand movement could be drawn by the motion because they did not follow the game carefully. This could be due to the fact that they are not paying full attention or are not fully aware of the rules.

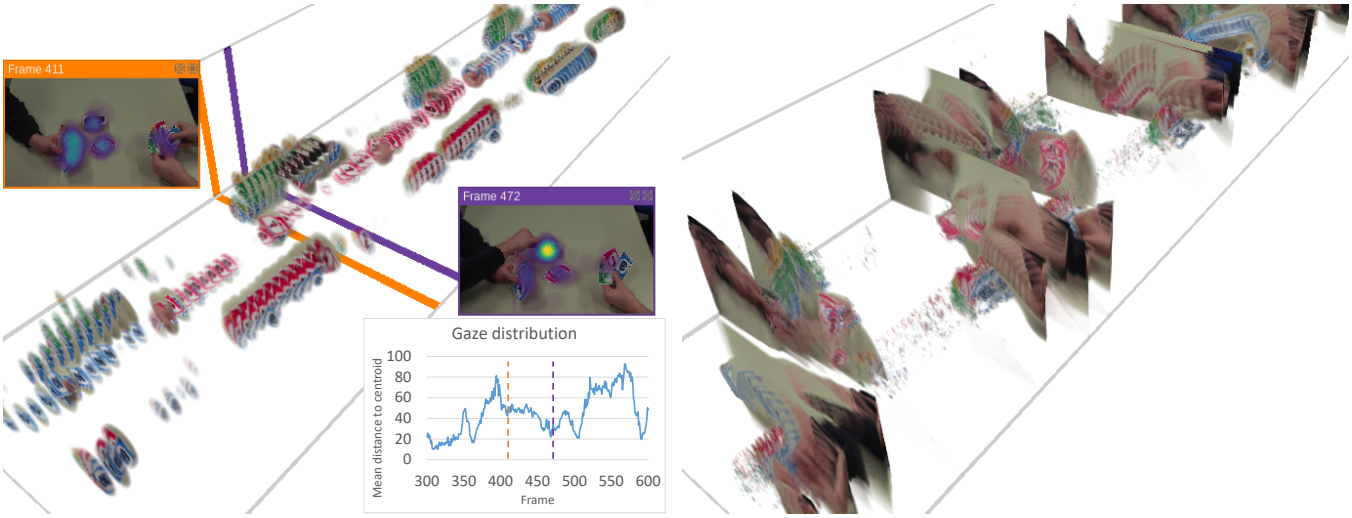
Our approach is also well suited to support statistical measures with illustrations. As an example, one can calculate the mean distance of gaze positions relative to the centroid over time, which is an indicator for attentional synchrony if values are low [22]. The stimulus context is not directly available if the mean distance alone is used. However, such a measure can be applied to segment respective time spans. With our STC approach, we complement the measure to directly depict what happend in the stimulus that caused the changes in the values. As an example, Figure 9 shows such a gaze distribution plot for a range around the selected frames.

5 DISCUSSION

Utility. The examples in Section 4 demonstrate the usefulness of our approach especially for gaining a combined overview of video, gaze, and flow data. In the first example, investigating the Kite video, our volume STC rendering applied with a gaze filter yields a concise overview of the participants' gaze distribution across the whole video. For example, the analyst can directly see how participants follow the kite, and that their focus shifts to the person steering the kite when it leaves the field of view. Furthermore, it is possible to make out details in our visualization that would be obscured by a traditional heat map. Starting at the overview, the analyst can easily pick single frames for further investigation or compare the gaze to the optical flow vectors, e.g., movement patters as also demonstrated in the second example (Thimblerig). The flow data also contains camera pans that clutter the visualizations of some of the videos (Car Pursuit and Kite). However, they could be automatically detected and removed with further pre-processing, which remains subject for future work. The UNO card game example in particular demonstrates how much detail can be shown in our STC visualization. A combination of filtering with gaze and flow data reveals major game moves as well as gaze patterns.

Filtering out specific angle ranges of the optical flow vector field did not reveal prominent patterns in the examples shown. However, we anticipate that this could be useful for analyzing other data sets, especially in combination with the filtering based on the magnitude of flow vectors. For instance, it could be used to filter for objects moving only in a certain direction such as a person walking from left to right while others move from right to left.

Comparison to Point Cloud Rendering. Our approach presents gaze patterns in an overview enriched with context from the visual stimulus. There are some advantages of this approach in comparison to point cloud rendering, a current state-of-the-art technique [21] (Figure 10). We evaluate both techniques based on three aspects an STC visualization for gaze data should be able to provide: (1) an



(a) Video data with low gaze filtered out. Frames before and of the first draw action are selected.

(b) Video data with low motion filtered out. The game moves of the two payers are clearly visible.

Figure 9: Two people playing the UNO card game visualized with our technique showing gaze and flow magnitude. The plot shows the mean distance of the gaze to the centroid for a selected time range. Selected frames are marked as vertical lines.

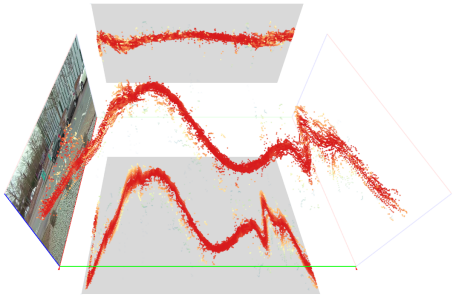


Figure 10: Point cloud rendering of gaze data based on the technique by Kurzhals et al. [21].

overview of common patterns, (2) the detection of outliers, and (3) information about the stimulus context.

In the overview, spatio-temporal patterns are clearly visible in both visualization techniques. The comparison between the point cloud and the volume is analogous to their 2D counterparts, the bee swarm and the heat map. The point cloud represents the data as it was recorded, including noise that might result from missing eye detection or an inaccuracy of the device. The aggregated volume displays the data temporally and spatially smoothed. Hence, noise is reduced and the analyst can focus on hot spots. Depending on the analysis task, the overview should enable an analyst to identify outlier behavior. For this task, the point cloud and the possibility to show scan paths by space-time trajectories is better suited than the volume. For temporal outliers, e.g., time spans of highly dispersive gaze distributions, the volume provides means to identify them efficiently. Kurzhals et al. [21] included a video plane in the STC that moves along the time axis. This approach provides context only through video skimming by adjusting a slider to a time step of interest. We incorporated this approach for annotation purposes, but with additional volume data from the video, it is possible to depict stimulus context in the overview without skimming.

In summary, a volume-based STC visualization of gaze and video data provides means for an overview of common spatio-temporal patterns in the data. In contrast to point clouds, it preserves the stimulus context and is therefore more suitable for efficient data exploration and the illustration of the results. To compensate for the search for outliers, a hybrid approach combining volume and gaze trajectories might be helpful.

6 CONCLUSION AND FUTURE WORK

We presented an approach to visualize gaze data from participants watching video. It shows the data as a space-time volume with multiple fields, and with this provides an overview of the stimulus context and how it relates to occurring gaze patterns. This helps an analyst identify important time spans without having to replay the whole video stimulus. The volume can further be explored flexibly and interactively via different transfer functions.

For future work, we plan to incorporate different approaches to facilitate the adjustment of transfer functions. Common presets for the identification of attentional synchrony, smooth pursuit of objects, or areas with high dispersion could help support common analysis tasks in eye tracking research. Furthermore, it is possible to convert additional information into spatio-temporal volumes. For example, pixel-precise AOI labels could be included by a numerical coding scheme. This would allow us to display only relevant regions when a specific AOI was visible.

ACKNOWLEDGEMENT

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Projektnummer 251654672 – TRR 161.

REFERENCES

[1] Johanna Beyer, Markus Hadwiger, and Hanspeter Pfister. 2015. State-of-the-Art in GPU-Based Large-Scale Volume Visualization. *Computer Graphics Forum* 34, 8 (2015), 13–37.

[2] Tanja Blascheck, Kuno Kurzhals, Michael Raschke, Michael Burch, Daniel Weiskopf, and Thomas Ertl. 2017. Visualization of Eye Tracking Data: A Taxonomy and Survey. *Computer Graphics Forum* 36, 8 (2017), 260–284.

[3] R. Borgo, M. Chen, B. Daubney, E. Grundy, G. Heidemann, B. Höferlin, M. Höferlin, H. Leitte, D. Weiskopf, and X. Xie. 2012. State of the Art Report on Video-Based Graphics and Video Visualization. *Computer Graphics Forum* 31, 8 (2012), 2450–2477.

[4] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. 2004. High Accuracy Optical Flow Estimation Based on a Theory for Warping. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 25–36.

[5] Valentin Bruder, Steffen Frey, and Thomas Ertl. 2017. Prediction-Based Load Balancing and Resolution Tuning for Interactive Volume Raycasting. *Visual Informatics* 1, 2 (2017), 106–117.

[6] Valentin Bruder, Marcel Hlawatsch, Steffen Frey, Michael Burch, Daniel Weiskopf, and Thomas Ertl. 2018. Volume-Based Large Dynamic Graph Analytics. In *Proceedings of the 22nd International Conference Information Visualisation (IV)*. 210–219.

[7] Min Chen, Ralf Botchen, Rudy Hashim, Daniel Weiskopf, Thomas Ertl, and Ian Thornton. 2006. Visual Signatures in Video Visualization. *IEEE Transactions on Visualization and Computer Graphics* 12, 5 (2006), 1093–1100.

[8] Robert A. Drebin, Loren Carpenter, and Pat Hanrahan. 1988. Volume Rendering. *SIGGRAPH Computer Graphics* 22, 4 (1988), 65–74.

[9] Klaus Engel, Markus Hadwiger, Joe M. Kniss, Christof Rezk-Salama, and Daniel Weiskopf. 2006. *Real-Time Volume Graphics*. A. K. Peters, Ltd., Natick, MA.

[10] Steffen Frey. 2018. Spatio-Temporal Contours from Deep Volume Raycasting. *Computer Graphics Forum* 37, 3 (2018), 513–524.

[11] Steffen Frey and Thomas Ertl. 2017. Flow-Based Temporal Selection for Interactive Volume Visualization. 36, 8 (2017), 153–165.

[12] Steffen Frey and Thomas Ertl. 2017. Progressive Direct Volume-to-Volume Transformation. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2017), 921–930.

[13] Steffen Frey, Filip Sadlo, and Thomas Ertl. 2012. Visualization of Temporal Similarity in Field Data. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (2012), 2023–2032.

[14] Peter Gatalsky, Natalia Andrienko, and Gennady Andrienko. 2004. Interactive Analysis of Event Data Using Space-Time Cube. In *Proceedings of the Eighth International Conference on Information Visualisation (IV)*. 145–152.

[15] Torsten Hägerstraand. 1970. What About People in Regional Science? *Papers in Regional Science* 24, 1 (1970), 7–24.

[16] Martin Hilpoltsteiner. 2005. *Recreating Movement – Tools for Analyzing Film Sequences*. Diplomarbeit. University of Applied Sciences, Wuerzburg, Germany.

[17] James T. Kajiya and Brian P. von Herzen. 1984. Ray Tracing Volume Densities. *SIGGRAPH Computer Graphics* 18, 3 (1984), 165–174.

[18] Maurice Koch, Kuno Kurzhals, and Daniel Weiskopf. 2018. Image-Based Scanpath Comparison with Slit-Scan Visualization. In *Proceedings of the ACM Symposium on Eye Tracking Research and Applications (ETRA)*. 55:1–55:5.

[19] Menno-Jan Kraak and Irma Kveladze. 2017. Narrative of the Annotated Space-Time Cube – Revisiting a Historical Event. *Journal of Maps* 13, 1 (2017), 56–61.

[20] Kuno Kurzhals, Cyrill Fabian Bopp, Jochen Bässler, Felix Ebinger, and Daniel Weiskopf. 2014. Benchmark Data for Evaluating Visualization and Analysis Techniques for Eye Tracking for Video Stimuli. In *Proceedings of the Workshop Beyond Time and Errors: Novel Evaluation Methods for Visualization (BELIV)*. 54–60.

[21] Kuno Kurzhals, Florian Heimerl, and Daniel Weiskopf. 2014. ISeeCube: Visual Analysis of Gaze Data for Video. In *Proceedings of the Symposium on Eye-Tracking Research and Applications (ETRA)*. 43–50.

[22] Kuno Kurzhals and Daniel Weiskopf. 2013. Space-Time Visual Analytics of Eye-Tracking Data for Dynamic Stimuli. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 2129–2138.

[23] Kuno Kurzhals and Daniel Weiskopf. 2016. Visualizing Eye Tracking Data with Gaze-Guided Slit-Scans. In *Proceedings of the IEEE Second Workshop on Eye Tracking and Visualization (ETVIS)*. 45–49.

[24] Xia Li, Arzu Çöltekin, and Menno-Jan Kraak. 2010. Visual Exploration of Eye Movement Data Using the Space-Time-Cube. In *Proceedings of the 6th International Conference on Geographic Information Science*. 295–309.

[25] Parag K. Mital, Tim J. Smith, Robin L. Hill, and John M. Henderson. 2011. Clustering of Gaze during Dynamic Scene Viewing Is Predicted by Motion. *Cognitive Computation* 3, 1 (2011), 5–24.

[26] Prithiviraj K. Muthumanickam, Katerina Vrotsou, Aida Nordman, Jimmy Johansson, and Matthew Cooper. 2019. Identification of Temporally Varying Areas of Interest in Long-Duration Eye-Tracking Data Sets. *IEEE Transactions on Visualization and Computer Graphics* 25, 1 (2019), 87–97.

[27] Cuong Nguyen, Yuzhen Niu, and Feng Liu. 2012. Video Summagator: An Interface for Video Summarization and Navigation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 647–650.

[28] Stefan Roettger, Stefan Guthe, Daniel Weiskopf, Thomas Ertl, and Wolfgang Strasser. 2003. Smart Hardware-accelerated Volume Rendering. In *Proceedings of the 2003 joint Eurographics/IEEE TCVG Symposium on Visualisation (VISSYM)*. 231–238.

[29] Mario Romero, Jay Summet, John Stasko, and Gregory Abowd. 2008. Viz-A-Vis: Toward Visualizing Video through Computer Vision. *IEEE Transactions on Visualization and Computer Graphics* 14, 6 (2008), 1261–1268.

[30] Bernard W Silverman. 1986. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, New York.

[31] Simon Stegmaier, Magnus Strengert, Thomas Klein, and Thomas Ertl. 2005. A Simple and Flexible Volume Rendering Framework for Graphics-Hardware-Based Raycasting. In *In Proceedings of the Fourth International Workshop on Volume Graphics*. 187–241.

[32] Jonathan Woodring and Han-Wei Shen. 2003. Chronovolumes: a Direct Rendering Technique for Visualizing Time-varying Data. In *Proceedings of the 2003 Eurographics/IEEE TCVG Workshop on Volume Graphics*. 27–34.