Visual Data Cleansing of Low-Level Eye Tracking Data

Christoph Schulz, Michael Burch, Fabian Beck, and Daniel Weiskopf

Abstract Analysis and visualization of eye movement data from eye tracking studies typically take into account gazes, fixations, and saccades of both eyes filtered and fused into a combined eye. Although this is a valid strategy, we argue that it is also worth investigating low-level eye tracking data prior to high-level analysis, because today's eye tracking systems measure and infer data from both eyes separately. In this work, we present an approach that supports visual analysis and cleansing of low-level time-varying data for eye tracking experiments. The visualization helps researchers get insights into the quality of the data in terms of its uncertainty, or reliability. We discuss uncertainty originating from eye tracking, and how to reveal it for visualization, using a comparative approach for disagreement between plots, and a density-based approach for accuracy in volume rendering. Finally, we illustrate the usefulness of our approach by applying it to eye movement data recorded with two state-of-the-art eye trackers.

1 Introduction

We start earlier than the typical process of eye tracking analysis and visualization, and argue that a separate visualization of low-level time-varying data can help explore the eye movements regarding reliability. Due to the wide variety of eye tracking experiments we introduce a generic reference workflow (Section 3). Our contributions are a discussion of how we can model uncertainty in the context of eye tracking (Section 4), a cleansing technique for time-series oriented eye tracking data (Section 5), and a visualization technique that reveals uncertainty of the left, right, and combined eyes (Section 6).

Christoph Schulz, Michael Burch, Fabian Beck, and Daniel Weiskopf
Visualization Research Center, University of Stuttgart, Allmandring 19, 70569 Stuttgart, Germany
e-mail: firstname.lastname@visus.uni-stuttgart.de

We demonstrate our cleansing approach and visualization technique by applying it to eye tracking data from previous eye tracking studies conducted with Tobii T60XL and SMI RED250 eye tracking devices (Section 8). As a major outcome, we find differences over time between left, right, and combined eyes while visually inferring credibility of the recorded data.

Furthermore, we provide our implementation under the terms of the MIT-License (Section 7).

2 Related Work

Much of the previous related work on eye tracking, data cleansing, and uncertainty occurs within isolated domains.

Eye Tracking Hardware: Singh and Singh [29] and Al-Rahayfeh and Faezipour [3] provide reviews of anatomical and technical aspects of eye tracking in general, which are used for discussion later on.

Eye Tracking Quality: Holmqvist et al. [18] note that standardized metrics would be of great help when assessing eye tracking data quality. They further argue that fixation filters and correlation with areas of interest may actually hide errors. Netzel et al. [23] increase fixation data quality through manual annotation of fixations. To our knowledge, it is much more common to enhance study quality by reducing measurement errors introduced by sampling frequency [4] or user movement [5, 12] than communicating uncertainty present in recorded data. In contrast, we specifically show the reliability of the data as a basis for user-controlled improvements of the data quality.

Eye Tracking Visualization: Eye movements recorded during eye tracking studies are typically analyzed and visualized by temporal aggregation like in attention maps [22]. While this allows us to derive hot spots [10] of visual attention, we cannot analyze time-varying patterns. If gaze plots [13] are used, the time-varying behavior is explicitly encoded in the visual representation, but for long-lasting tasks and a larger number of study participants, the amount of visual clutter increases, making such a visualization difficult to read. Many visualization techniques have already been developed to analyze eye movement data for patterns [9], but most of them only take aggregated eye movements into account. Hence, we base our work on simple line plots, scarf plots [25], and space-time cubes [21]. We furthermore focus on including uncertainty visualization, which has largely been ignored in eye tracking visualization so far.

Time-Series Visualization: Eye tracking data is sampled and time-dependent, so we would like to point out a variety of techniques to visualize different aspects of time [2]. Stacking representations of data were previously used by Shahar et al. in their tool called KNAVE-II [28] in the clinical domain with a focus on semantic navigation. VisuExplore [26] and CareCruiser [15] also use stacked representations of time series while focusing on medical use cases and structure of patient data. Beard et al. [7] describe a system to explore spatial and temporal patterns of sensor data,

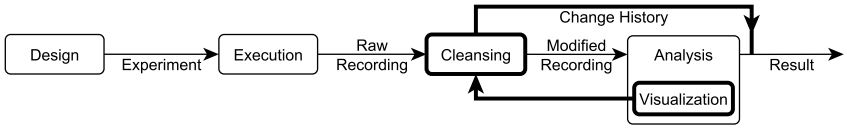

Fig. 1 Reference workflow for eye tracking experiments. Experiments are designed and executed to gain raw recordings, which can be cleansed and analyzed to obtain a result. The final results of the study are influenced also by how the cleansing was performed.

while dealing with uncertainty of missing data to some extent. We transfer the idea of stacked representations to eye tracking by including specialized representations for eye tracking data.

Uncertainty Visualization: We apply a rationale by Skeels et al. [30] to eye tracking, stating that visualizing uncertainty could help make better decisions. Furthermore, we base our discussion on a review of uncertainty visualization by Brodlie et al. [11]. We distinguish between an accuracy-based and a comparative approach in terms of uncertainty.

Data Cleansing: Rahm and Do [24] classify data quality problems for data cleansing in the data warehouse domain. Their definition of single-source and multi-source problems transfers well to our work. Kandel et al. [20] describe a technique to interactively infer mapping functions from manipulation of data, but they do not deal with sequential, time-varying data. Gschwandtner et al. [16] propose design principles and techniques to exploit time specifics for data cleansing, but they do not visualize or propagate different facets of uncertainty originating from a processing pipeline, such as eye tracking.

3 Experiment Workflow

Before discussing uncertainty in eye tracking, we elaborate on the integration of our tool into a reference experiment workflow, shown in Figure 1: An eye tracking experiment is designed by a researcher and executed to obtain a recording of time-varying data for each participant. It may be very hard to analyze raw recordings, because recordings may require segmentation and re-ordering, contain recognition errors, and originate from multiple eye trackers. Therefore, proper and comprehensible data cleansing prior to analytics can reduce data quality issues by identifying corrupt data and making data consistent. Our approach to data cleansing is iterative and based on visual feedback. Additionally, a description of how the cleansing was done should be incorporated into the final results, because cleansing has the same potential to hide errors, as fixation filters do [18].

Considering the range of available eye trackers and different types of experiments, we made as few assumptions as possible about hardware and use cases to

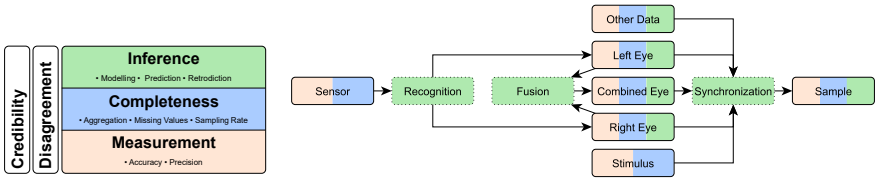

Fig. 2 (a) Classification of uncertainty and (b) eye tracking pipeline, introducing and propagating different kinds of uncertainty. Each eye is recognized individually and then fused to a combined eye. Subsequently, all data is synchronized and emitted as samples.

keep our approach generic. We assume that data is a time-dependent series of samples and for implementation reasons, we limit ourselves to stationary eye tracking setups using static images and video stimuli. We tested a maximum duration of about half an hour per recording, even though our implementation should perform well beyond that.

4 Modeling Uncertainty in Eye Tracking Data

Our model is based on one question: What reveals flawed data and separates it from trustworthy data, and thus is crucial for decision making during cleansing? This question leads to the topic of uncertainty, which introduces data analysis challenges that we will discuss in the following sections. We adhere to the term uncertainty of the visualization community. Other communities prefer data confidence, quality, or trust.

4.1 Background

We adopt a classification by Skeels et al. [30] to discuss different aspects of uncertainty in the context of eye tracking. Their classification distinguishes between measurement, completeness, inference, disagreement, and credibility (see Figure 2a). Measurement uncertainty describes accuracy and precision. Completeness uncertainty describes aggregation, missing values, and sampling. Inference uncertainty describes modeling, prediction, and retrodiction. The former three levels are stacked because uncertainty propagates from bottom to top (right part of Figure 2a), whereas disagreement and credibility on the left span all three levels as derived characteristics (left part of Figure 2a).

We apply this classification to a simplified eye tracking pipeline, condensed from related work [3, 27, 29, 32] and depicted in Figure 2b to illustrate sources of uncertainty originating from eye tracking. The process of optical eye tracking starts with

Table 1 Measurement-related technical data extracted from vendor documentation, listing upper bounds in case of doubt.

Device	Sampling Rate [Hz]	Accuracy [°]	Latency [ms]
Tobii			
T60	60	0.5	33
T120	120	0.5	33
SMI			
REDn Scientific	30 or 60	0.5	25
RED 250 mobile	60, 120, or 250	0.4	8
RED 500	500	0.4	4
Eye Tracking Glasses 2.0	30 or 60	0.5	measured

light hitting a raster of sensor pixels, aggregated to a sequence of images, forming a video. This process introduces measurement uncertainty, because of physical properties such as lenses, pixel density, and signal-to-noise ratio, and completeness uncertainty, because of missing eye movements due to a low sampling rate, and the light conditions might be too bad for the pixels to work properly. Subsequently, inference uncertainty is introduced, as each eye is recognized independently, fused into a combined eye, and synchronized with other data, e.g., keyboard and mouse events composing a sample.

Many eye trackers address uncertainty algorithmically, e.g., internal latencies get canceled out, and missing values are estimated using a co-simulation of the participant's eyes [33]. Most vendors provide uncertainty-related information as a part of technical specifications and recorded data, e.g., angular gaze accuracy, sampling resolution, and recognition confidence. Hence, an eye tracking device exhibits all three levels of uncertainty and many of its sub-systems have to be considered as black boxes. Unfortunately, this means that we have to rely on information provided by vendors, which limits our basis for revealing uncertainty.

As examples, we have examined several Tobii and SMI eye tracking systems, shown in Table 1 and Table 2. They provide quite different quality metrics and technical specifications: Tobii defines a validity code to represent the success of recognition, whereas SMI emits several Boolean and ordinal values for conveying recognition confidence and timing issues with device-dependent availability. With our generic model and handling of uncertainty, we aim to cover this whole variety of technology-driven descriptions of data uncertainty.

We believe that visualizing uncertainty helps researchers find disagreement in data and estimate a credibility of their recordings. Formally, we model and visualize two different types of uncertainty: An accuracy and precision uncertainty model that is based on probability density functions (PDFs) for spatial and temporal dimensions of data, and a failure uncertainty model that is based on black-box metrics emitted by the eye tracker, that indicate whether the data sample was invalidated (Table 2). Visualizing these models allows us to distinguish during cleansing between measurement-related issues, recognition failures, and ambiguities.

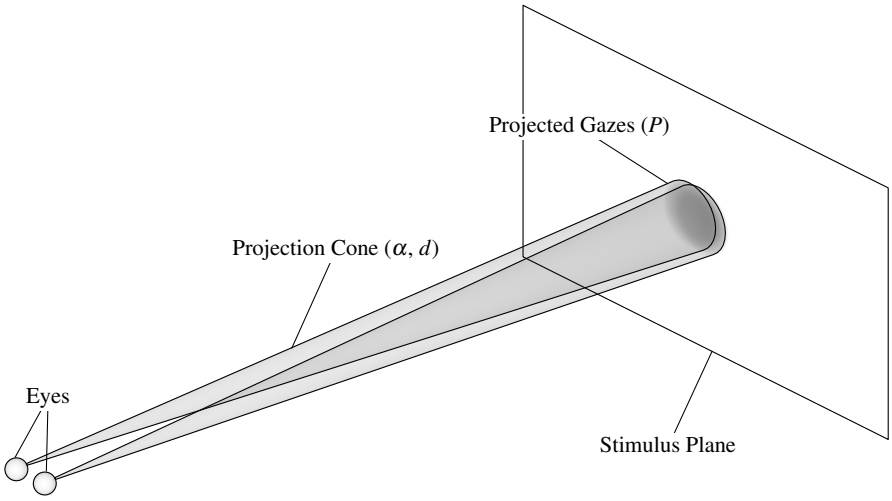

Fig. 3 Projection from eye space to gaze space using cones. Measurement imprecision and disagreement between the left and right eye are exaggerated for illustration purposes.

4.2 Gaze Data

We focus on uncertainty present in gaze positions for the eye $P \subset \mathbb{R}^2$, and time $T \subset \mathbb{R}$, modeled as PDF. For one sample, the PDF reads:

$$\rho_P \colon T \times P \to [0,1] \tag{1}$$

Here ρ_P describes the probability of a gaze position to take a given value.

Visually, we aim at propagation of uncertainty by expanding every projection line to a cone (Figure 3). Note that this cone does not model aspects of the human visual system, such as foveal acuity or gaze contingency. This is only about transforming accuracy and precision of the eye tracker to gaze space, i.e., we need to have all data in gaze space. Since the angular accuracy σ_P is not directly provided in gaze space, we estimate it using σ_α provided in eye space (Table 1) and the eye–stimulus distance d:

$$\sigma_P = d \ \sin \sigma_\alpha \tag{2}$$

When exploring the density-based visualization shown later on, one should be aware how d was determined, i.e., manually measured or by guessing, because the impression of accuracy and precision can be deceiving otherwise. Angular accuracy is the proximity of measured samples to the true direction the eye is looking. This is not to be confused with precision, denoting reproducibility of the measurement. Hence, we argue that a normal distribution with 2σ should be assumed, because it covers 95% its area. This allows us to define a PDF for each gaze position $\mathbf{p} = (p_x, p_y)$ and accuracy $\sigma_\mathbf{P} = (\sigma_x, \sigma_y)$:

Table 2 Quality metrics recorded by vendor software.

Field	Description	Scale
Tobii Studio		
Validity(Left|Right)	Validity code for each eye	nominal
SMI iView		
(L|R) Validity	General quality value for each eye	Boolean
Pupil Confidence	Validity of the pupil diameter	Boolean
Timing	Indicates a timing violation	Boolean
Latency	Required time to process a sample	μs

$$\varphi(\mathbf{p};\sigma_{\mathbf{P}}) = \varphi_0 e^{-\left(\frac{(p_x(2/\sigma_x))^2}{2}+\frac{(p_y(2/\sigma_y))^2}{2}\right)} \tag{3}$$

Here we assume that the sample position at the origin and normalization using φ_0.

Time can me modeled as simple box function at time t, and sample intervals Δt, if we assume a uniform probability distribution within a temporal sampling interval:

$$\xi(t;\Delta t) = \frac{1}{\Delta t}\begin{cases}1 & \text{if } t \in [-\Delta t/2, \Delta t/2] \\ 0 & \text{otherwise}\end{cases} \tag{4}$$

Again the sample time is at the origin.

We assume that the probability densities for spatial position, φ, and time, ξ, are independent. Therefore, the overall probability ρ_P, is obtained by multiplication:

$$\rho_P(t,\mathbf{p};\Delta t,\sigma) = \varphi(\mathbf{p};\sigma)\xi(t;\Delta t) \tag{5}$$

This could be easily extended to binocular vision by adding ρ_P of the left and right eye.

In addition to accuracy and precision issues, the eye tracker can invalidate the sample, which is denoted by metrics, shown in Table 2. Hence, the PDF is only valid if the eye tracker did not invalidate the sample. Invalid samples have to be treated separately by any subsequent processing, in particular by our visualization techniques.

If it is not possible to determine an eye–stimulus transformation, as in Figure 3, because the required depth information is missing, the depth value needs to be approximated. Most eye trackers make an educated guess by defining the recorded video as stimulus, because the camera–eye distance can be measured quite well. Although this is a good approximation, it poses problems when dealing with multiple recordings, because finding a common space can be a hard problem of itself, especially for mobile devices. Fortunately, it is easy to solve for stationary devices, which allows combined cleansing and preliminary analysis of multiple recordings.

4.3 Signal and Event Data

A recording may include other data that helps put gaze data into context. Signals $S \subset \mathbb{R}^n$, such as electroencephalography (EEG), motion capturing, or eye tracker latencies, can be represented as time-series data analogously to gaze data. Again, we use a PDF to describe uncertainty:

$$\rho_S \colon T \times S \to [0,1] \tag{6}$$

Events with parameters $E \times P$, such as keyboard, mouse, and touch input, are considered discrete. We neglect domain-specific uncertainty models for events, e.g., a model for accidental keyboard strokes, and thus only time T may be augmented by a PDF, i.e.:

$$\rho_E \colon T \to [0,1] \tag{7}$$

Essentially, this is a generalization of our uncertainty model to fit all remaining aspects of a recording. For the sake of simplicity, we do not deal with manually annotated events [8].

5 Cleansing Technique

Our processing model is based on the pipes and filters pattern, also used by ParaView [1] and VisTrails [6]. Data is interpreted as immutable and processed by a pipeline composed of functions. A function can perform any non-destructive mapping of data. Formally, we want to setup a processing graph $G = (F,C)$ without cycles composed of functions F, a sink function $s \in F$, and connections C. If data is pushed or pulled, all affected functions are recomputed according to their dependencies in the graph. We have chosen this design because of its flexibility. Uncertain and certain data are treated as values V, i.e.:

$$f \colon T \times V_1 \times \ldots \times V_n \to T \times V_1 \times \ldots \times V_m \tag{8}$$

Visual cleansing means inspecting a function's visual response, while adjusting (optional) parameters to manipulate data. To illustrate this concept, we describe a couple of use cases and functions:

Velocity, Acceleration, and Jerk might be of interest in general for any time-varying positional data. All values can be obtained by simply chaining a differential operator up to three times.

Filtering is likely to be useful for repairing corrupt data. Such a function could do interpolation if a trigger signal is set, and pass-through otherwise. Another approach would be to reconstruct corrupt data using approximation.

Let us assume a simple cleansing function that drops data if a trigger signal is set and does pass-through otherwise. Depending on the amount of data gathered, tuning

this trigger signal can be a tedious task. We spare users from small-scale work, by the following approach: We employ trigger signal emitting functions that analyze values, such as recognition confidences or other quality metrics, which can then be fine-tuned and used for other cleansing functions. Assuming the processing graph was set up correctly, this approach allows going from macroscopic to microscopic cleansing iteratively and quickly.

6 Visualization Technique

Our technique uses a stack of specialized, time-aligned visualizations. Time runs from left to right and the stack is sorted in a user-defined fashion. Plot-based stack elements reveal gaps in the data, disagreement between the individual and combined eyes, i.e., the failure uncertainty model. Volume-based stack elements reveal measurement imprecision, i.e., the accuracy and precision uncertainty model. In the following sections, we will discuss how uncertainty connects cleansing and visualization.

6.1 Stereo Plot

We use stereo plots for eye-related data—the name originates from the fact that the right eye plot is flipped below the left eye plot and the combined eye is rendered on top of both. Stereo plots manifest as line plots and scarf plots [25]. Of course, they may be used without this little stereo twist as well, like shown later on. In terms of our uncertainty model, stereo plots visualize the failure uncertainty model through comparison—the accuracy and precision uncertainty model is omitted here.

The line variant is used for ratio-scale data, as shown in Figure 4. Differences between the individual eyes can be inferred by comparing the top plot Ⓛ and bottom plot Ⓡ. Both plots are overlaid with thick lines Ⓒ, representing the combined eye. Undefined samples originating from the eye tracker or cleansing are depicted as gaps shown at ①. Disagreement can be observed at ②.

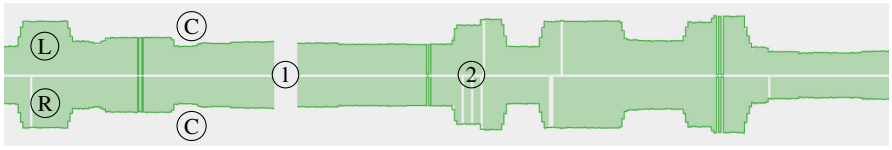

Fig. 4 A stereo line plot showing the left, right, and combined eye (gaze position). The right eye chart is flipped below the left eye plot and the combined eye plot is drawn as overlay of the top and bottom plot.

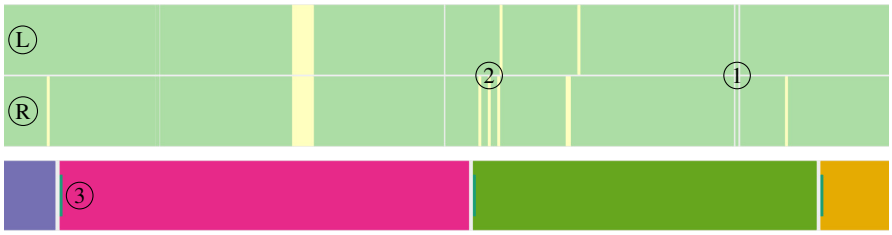

Fig. 5 Top: A stereo scarf plots, showing validity over time (green: good, otherwise: error) for the left and right eye. The right eye chart is flipped below the left eye plot. Bottom: A scarf plot showing activity (big bars: stimulus visibility, small bars: user input)

The scarf plot variant is used for nominal data such as validity codes and events, as shown in Figure 5. Again, differences can be inferred by comparing the top scarf plot Ⓛ and bottom scarf plot Ⓡ. The combined eye is missing because there was no nominal data for the combined eye. Undefined samples are shown at ①, disagreement is shown at ②. We also use non-stereo scarf plots to depict concurrent activities, hence a scarf plot may be subdivided vertically to indicate concurrency, depicted at ③. We use ColorBrewer palettes [17] to encode quantities by color.

6.2 Space-Time Cube

We visualize gaze positions using a space-time cube, as shown in Figure 6, containing a video plane Ⓐ, similar to the one by Kurzhals and Weiskopf [21], except that we use an orthographic instead of a perspective projection and volumes Ⓑ instead of solid points for rendering. The former prevents distortions around a vanishing point at the cost of a natural feeling of depth. The latter allows us to encode more information such as uncertainty. Hence, our space-time cube provides a visual impression of the accuracy and precision uncertainty model using density, i.e., the failure uncertainty model is neglected. Low density equals high uncertainty. The video plane provides some additional context during cleansing. At ① magic values emitted by the eye tracker can be observed, while Ⓑ indicates fixations. This would be hard to see if gaze positions are depicted using solid points of arbitrary size. Remember that we use a box function to visualize time and a Gaussian function for spatial precision and accuracy, which allows us to do proper scaling of time and space.

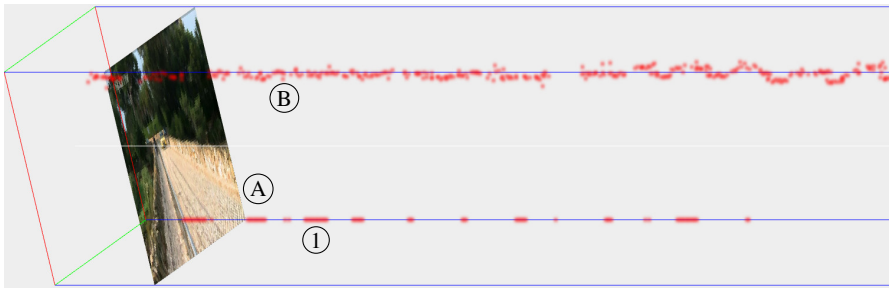

Fig. 6 An orthographic space-time cube showing gaze positions over time (from left to right), augmented by a PDF along with a video plane to provide some additional context.

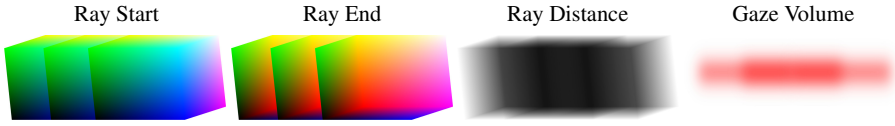

Fig. 7 Volume rendering of three gazes. The bounding box is intersected with the camera-ray in the fragment shader (ray start, ray end, and ray distance). Afterward, ray marching is applied to compute the gaze volume. All densities are blended together afterward.

7 Implementation

We provide our C++/Qt-based implementation under the terms of the MIT-License on GitHub[1].

The pipes and filters pattern maps directly to code, e.g., data is loaded by a *LoadCSV* class, passed to cleansing function classes, and the visualizations are managed by *Display* classes. In terms of rendering, we took two different approaches. Stereo plots and other simple elements are rendered using the *QPainter* API, while space-time cubes are rendered using native OpenGL.

We had to mix a couple of rendering approaches to achieve crisp images and decent performance for our space-time cube. The basic idea is splatting of ray-casted volumes. Splatting is done additive, as shown by Hopf et al. [19]. Each sample is ray cast, as done by Stegmaier et al. [31]. This is different from what Djurcilov et al. [14] do for volume rendering of uncertainty, i.e., their approach of storing the density information inside a volume texture is not feasible for sparse data, such as gaze positions, because of memory constraints.

Gaze points are passed as vertex attributes to the shader program. All points are transformed in the vertex shader and expanded to a cube in the geometry shader. Afterward, we apply ray marching and splatting, shown in Figure 7. This allows us to render the gaze positions using one draw call, resulting in decent performance. Note that the video plane and axes are rendered using separate draw calls.

[1] https://github.com/schulzch/BinocularVis

8 Case Study

The case study aims at showing how to apply the visualization technique, i.e., how to identify typical relationships between dimensions of eye tracking data that indicate an error. We have used two data sets for this case study.

The first data set is from a study with five participants using a Tobii T60XL and Tobii Studio 2.2.8. The test was conducted in a distraction-free room, illuminated with diffuse light. The participants had to match a line to a pair of dots with varying line lengths and point distances, leading to very fast, comparative eye movements. The second data set is from a study with fifteen participants using an SMI RED250 and iView 2.8.26. The test was conducted under similar conditions. The participants had to interact with small button-sized input elements of high information density, depicting a probability density function, leading to very subtle gaze position changes. We consider both studies as typical representatives and potentially vulnerable to accuracy, precision, and missing data issues. We start with a description of all visual representations of eye tracking data used in our case study:

Time is represented as simple ruler and measured in seconds.

Activity is represented as scarf plot. Long bars depict the duration of image stimuli, i.e., their visibility. Short bars (at the end of stimuli) depict mouse clicks by the participant.

Validity is represented as stereo scarf plot for each eye individually. Combined eye confidence is not emitted by the eye tracker. Light green depicts "all fine", other colors depict "error".

Processing Latency is represented as non-stereo line plot and measured in microseconds. SMI devices start to drop samples if latency is too high.

Camera–Eye Distance & Pupil Sizes are represented as stereo line plots. The former is the Euclidean distance computed from the eye coordinates. The latter is an estimate of the true pupil's size.

Gazes are represented as space-time cube containing density splats and a video plane for additional context.

Gaze X & Y are represented as stereo line plots. In addition to the individual eyes, the combined eye is depicted using a darkened line.

Gaze Velocity is derived from gaze coordinates through differentiation and represented as stereo line plot. Again, the combined eye is depicted using a darkened line.

We will demonstrate cleansing by visually inspecting slices Ⓐ to Ⓓ from the first data set shown in Figure 8 and draft possible steps for cleansing.

Slice Ⓐ shows minor recognition jitter for both eyes in the validity plot around 1.5 s, which reveals corrupt values in the plots below. Note how the dark lines of the combined eye in the gaze plots do not intersect with the light colored areas of the individual eye data. This allows us to infer that the eye tracker has repaired those errors during eye fusion. If one wants to compare the left, right, and combined eye data in an analysis, it might be a good idea to reconstruct the missing individual eye data, but for this data set it does not matter.

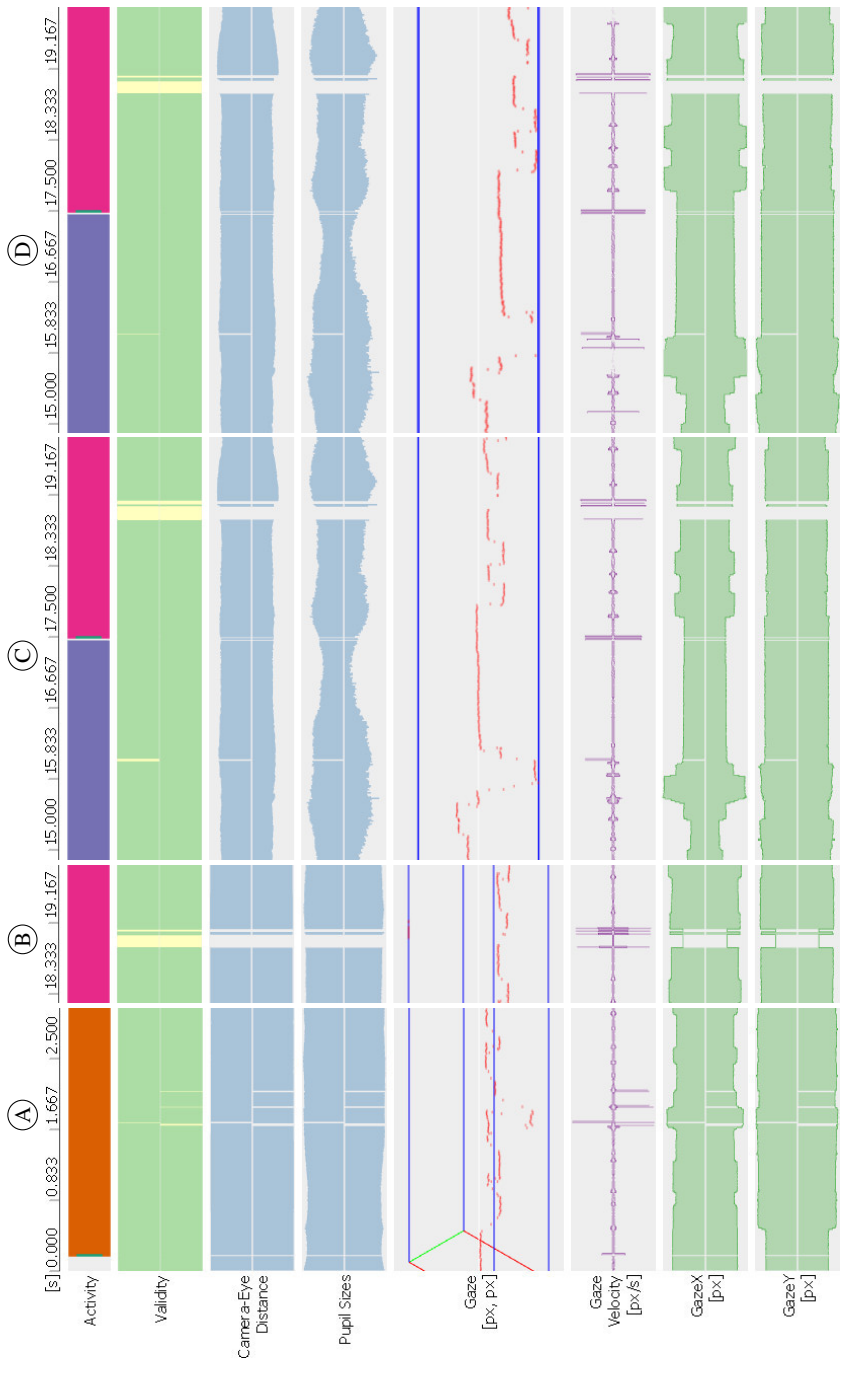

Fig. 8 Four slices from our first data set. From top to bottom: time, activity, validity, camera–eye distance, pupil size, gaze coordinates, gaze velocity, and decomposed gaze coordinates.

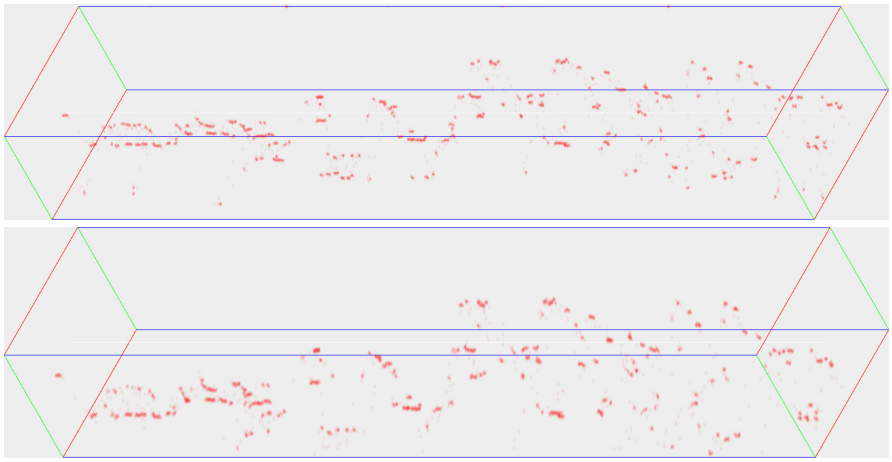

Fig. 9 Non-cleansed (top) and cleansed (bottom) space-time cubes from our first data set. Note dropped gazes close to the boundaries, where the participant was looking off-screen.

Slice Ⓑ reveals a real issue, as recognition of both eyes failed around 19 s. We can safely assume that the eye tracker does not use simple interpolation to fill in missing data, since the small time segment between the recognition errors does not seem to be a supporting point. After inspecting actual values, it turns out that the eye tracker emits zero for combined eye data and a negative magic value for individual eye data. This is documented behavior, even though it is considered bad practice to encode undefined and other magic values in $\mathbb{R}$. In addition, notice how camera–eye distance plot and pupil size plot seem featureless, because of the same issue. We fix this issue by filtering the corresponding data, i.e., setting magic values to undefined.

Slice Ⓒ shows the same data with cleansing functions applied. The pupil size plot and camera–eye distance plot now reveal many more features because scaling is no longer influenced by magic values. While rotating the space-time cube and correlating data with the stimulus image (not shown), we notice a weird offset in the data around 5.7 s. After inspection of the actual values, we know that the participant was looking off-screen. We decide to clamp the individual eye data to the screen resolution and set the combined eye data to undefined, if the participant was looking off-screen, so it will not interfere with post-cleansing analysis.

Slice Ⓓ shows the same data with more cleansing functions applied. The missing line on top of the lower two stereo line plots indicates that the participant was looking off-screen. Further inspection of the data reveals that this happens a couple of times during comparative eye movements, indicating that this could be an issue resulting from the combination of the stimuli and eye tracker. Just to make sure we removed the data in question using a filter-and-drop function (not shown).

An interesting observation across slices Ⓒ and Ⓓ is that the participant's pupil sizes seem to be unsynchronized. Unless this is caused by the eye tracker, it might be worth investigating.

We finish cleansing of this participant's data set by comparing the space-time cube's pre- and post-cleansing, shown in Figure 9. This also gives us a first impression of where fixations are (areas of high density).

The first data set is about 45 to 90 seconds per participant. It took a couple of minutes to cleanse data from the first participant. Data of other participants was much faster to cleanse.

The second data set is about 15 minutes per participant. In addition to that, the eye tracker has a high sampling rate, hence we choose a different strategy: take a participant's data with minor modifications, or not. We demonstrate this strategy by visually inspecting about 4 minutes (one task) from one participant, shown in Figure 10. We can confirm that the participant was reading instructions between Ⓐ and Ⓑ, because the *start* button is in the center of the screen and a line-reading pattern is revealed though density. This can be easily confirmed using the video plane. Around Ⓒ, the behavior changes much, the latency peaks, and the eye tracker starts to drop data. Using the video plane, we can confirm that the participant has finished the task Again, through inspection of the video plane, the participant has finished the task, hence interesting eye movement data is between Ⓑ and Ⓒ. At Ⓓ, a latency peak in the region of interest and missing data can be observed. Through inspection of the video plane and raw values, the source of this issue is unclear. Because it did not interfere much with the experiment, we drop the affected samples. Around Ⓔ, recognition of the right eye seems odd, but gaze positions look fine, so we decide to ignore it. Finally, we cut the region of interest from the data set. This process took less than a minute to complete.

9 Conclusion and Future Work

We discussed a two-layered uncertainty model in the context of eye tracking and a processing model for data cleansing. Additionally, we presented a technique to visually deduce disagreement and credibility by comparing time-aligned, stacked representations of eye tracking data. In particular, comparing the left and right eye against the combined eye seems to be a good strategy. Results show that data cleansing can be a fast process, given that enough visual context is provided, such as raw values, a stimulus view, and measured data augmented by uncertainty. We have increased data quality by dropping and clamping unexpected samples. Additionally, we provide an open source implementation for the interested reader.

In future work, want to extend our density-based approach to areas of interest and fixation filters. We believe that fuzzy intersection with areas of interest might be useful to convey and increase trust into data during analysis. From our own experience, fixation filters are sometimes difficult to tune, hence we envision sensitivity analysis for fixation filters, by mapping the parameter space of fixation filters to gaze space using volume rendering.

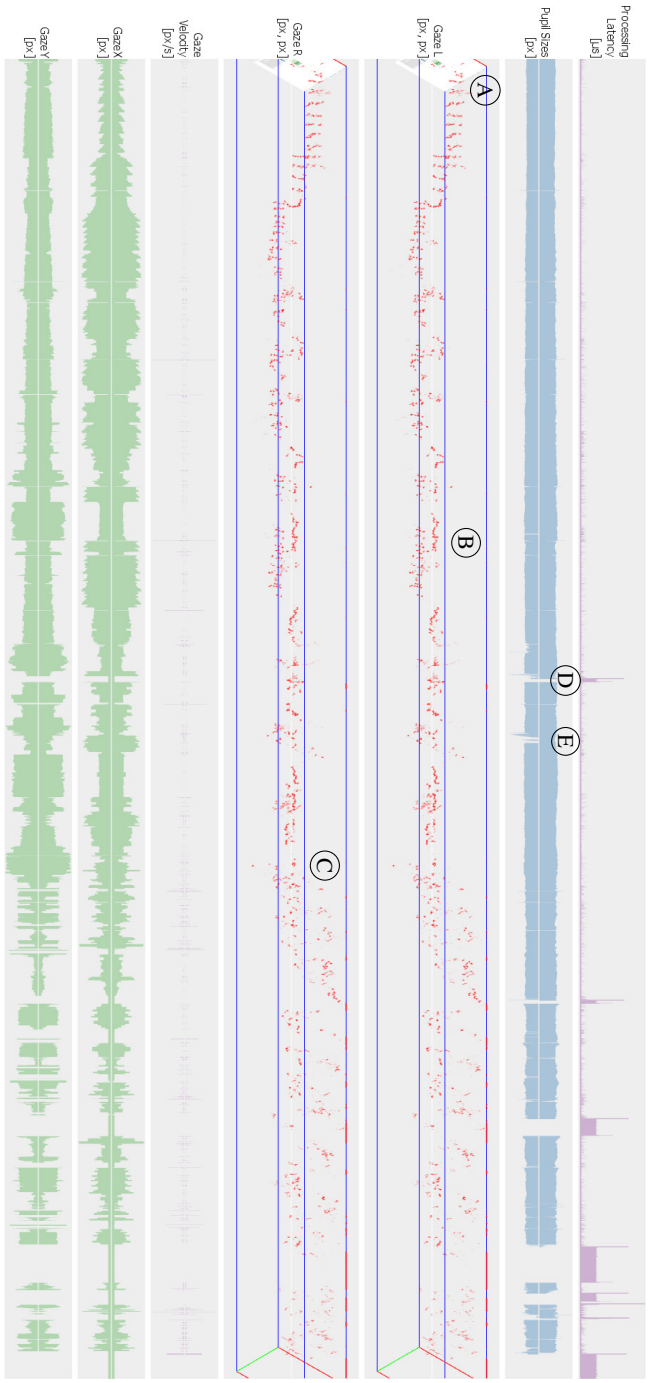

Fig. 10 A high-level view from our second data set, one participant. From top to down: processing latency, pupil sizes, gazes of the left eye, gazes of the right eye, along with gaze velocity and decomposed gaze coordinates of both eyes.

Acknowledgements We would like to thank the German Research Foundation (DFG) for financial support within project A01 of SFB/Transregio 161.

References

1. Ahrens, J., Geveci, B., Law, C.: ParaView : An end-user tool for large data visualization. Energy **836**, 717–732 (2005)
2. Aigner, W., Miksch, S., Müller, W., Schumann, H., Tominski, C.: Visualizing time-oriented data—a systematic view. Computers & Graphics **31**(3), 401–409 (2007)
3. Al-Rahayfeh, A., Faezipour, M.: Eye tracking and head movement detection: A state-of-art survey. Translational Engineering in Health and Medicine **1** (2013)
4. Andersson, R., Nyström, M., Holmqvist, K.: Sampling frequency and eye-tracking measures: how speed affects durations, latencies, and more. Journal of Eye Movement Research **3**(3), 1–12 (2010)
5. Barz, M., Bulling, A., Daiber, F.: Computational modelling and prediction of gaze estimation error for head-mounted eye trackers (2015)
6. Bavoil, L., Callahan, S.P., Crossno, P.J., Freire, J., Scheidegger, C.E., Silva, T., Vo, H.T.: VisTrails: Enabling interactive multiple-view visualizations. Proceedings of IEEE Visualization pp. 135–142 (2005)
7. Beard, K., Deese, H., Pettigrew, N.R.: A framework for visualization and exploration of events. Information Visualization **7**(2), 133–151 (2007)
8. Blascheck, T., John, M., Koch, S., Kurzhals, K., Ertl, T.: VA^2: A visual analytics approach for evaluating visual analytics applications. IEEE Transactions on Visualization and Computer Graphics **22**(1), 61–70 (2016)
9. Blascheck, T., Kurzhals, K., Raschke, M., Burch, M., Weiskopf, D., Ertl, T.: State-of-the-art of visualization for eye tracking data. In: EuroVis STAR, pp. 63–82 (2014)
10. Bojko, A.: Informative or misleading? Heatmaps deconstructed. In: J. Jacko (ed.) Human-Computer Interaction. New Trends, *Lecture Notes in Computer Science*, vol. 5610, pp. 30–39. Springer Berlin Heidelberg (2009)
11. Brodlie, K., Allendes Osorio, R., Lopes, A.: Expanding the Frontiers of Visual Analytics and Visualization, chap. A Review of Uncertainty in Data Visualization, pp. 81–109. Springer London (2012)
12. Cerrolaza, J.J., Villanueva, A., Villanueva, M., Cabeza, R.: Error characterization and compensation in eye tracking systems. In: Proceedings of the Symposium on Eye Tracking Research and Applications, ETRA '12, pp. 205–208 (2012)
13. Çöltekin, A., Fabrikant, S., Lacayo, M.: Exploring the efficiency of users' visual analytics strategies based on sequence analysis of eye movement recordings. International Journal of Geographical Information Science **24**(10), 1559–1575 (2010)
14. Djurcilov, S., Kim, K., Lermusiaux, P.F.J., Pang, A.: Volume rendering data with uncertainty information. Proceedings of the Joint EUROGRAPHICS and IEEE TCVG Symposium on Visualization pp. 243–252 (2001)
15. Gschwandtner, T., Aigner, W., Kaiser, K., Miksch, S., Seyfang, A.: CareCruiser: Exploring and visualizing plans, events, and effects interactively. Proceedings of IEEE Pacific Visualization Symposium, PacificVis pp. 43–50 (2011)
16. Gschwandtner, T., Aigner, W., Miksch, S., Gärtner, J., Kriglstein, S., Pohl, M., Suchy, N.: TimeCleanser: A visual analytics approach for data cleansing of time-oriented data. In: Proceedings of the 14th International Conference on Knowledge Technologies and Data-driven Business, i-KNOW '14, pp. 18:1–18:8 (2014)
17. Harrower, M., Brewer, C.A.: ColorBrewer.org: an online tool for selecting colour schemes for maps. The Cartographic Journal **40**(1), 27–37 (2003)

18. Holmqvist, K., Nyström, M., Mulvey, F.: Eye tracker data quality: What it is and how to measure it. In: Proceedings of the Symposium on Eye Tracking Research and Applications, ETRA '12, pp. 45–52 (2012)
19. Hopf, M., Luttenberger Michael, M., Thomas, E.: Hierarchical splatting of scattered 4D data. IEEE Computer Graphics and Applications **24**(4), 64–72 (2004)
20. Kandel, S., Paepcke, A., Hellerstein, J., Heer, J.: Wrangler: Interactive visual specification of data transformation scripts. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 3363–3372 (2011)
21. Kurzhals, K., Weiskopf, D.: Space-time visual analytics of eye-tracking data for dynamic stimuli. IEEE Transactions on Visualization and Computer Graphics **19**(12), 2129–2138 (2013)
22. Mackworth, J.F., Mackworth, N.H.: Eye fixations recorded on changing visual scenes by the television eye-marker. Journal of the Optical Society of America **48**(7), 439–445 (1958)
23. Netzel, R., Burch, M., Weiskopf, D.: Interactive scanpath-oriented annotation of fixations. In: Proceedings of the Symposium on Eye Tracking Research and Applications, ETRA '16, pp. 183–187 (2016)
24. Rahm, E., Do, H.H.: Data cleaning: Problems and current approaches. IEEE Data Engineering Bulletin **23**(4), 3–13 (2000)
25. Richardson, D.C., Dale, R.: Looking to understand: The coupling between speakers' and listeners' eye movements and its relationship to discourse comprehension. Cognitive Science **29**(6), 1045–1060 (2005)
26. Rind, A., Aigner, W., Miksch, S., Wiltner, S., Pohl, M., Turic, T., Drexler, F.: Visual exploration of time-oriented patient data for chronic diseases: Design study and evaluation. In: Information Quality in e-Health. Lecture Notes in Computer Science 7058, pp. 301–320 (2011)
27. SensoMotoric Instruments GmbH: BeGaze 2.4 Manual (2010)
28. Shahar, Y., Goren-Bar, D., Boaz, D., Tahan, G.: Distributed, intelligent, interactive visualization and exploration of time-oriented clinical data and their abstractions. Artificial Intelligence in Medicine **38**(2), 115–135 (2006)
29. Singh, H., Singh, J.: Human eye tracking and related issues: A review. International Journal of Scientific and Research Publications **2**, 1–9 (2012)
30. Skeels, M., Lee, B., Smith, G., Robertson, G.G.: Revealing uncertainty for information visualization. Information Visualization **9**(1), 70–81 (2010)
31. Stegmaier, S., Strengert, M., Klein, T., Ertl, T.: A simple and flexible volume rendering framework for graphics-hardware-based raycasting. In: Proceedings of the Fourth Eurographics / IEEE VGTC Conference on Volume Graphics, VG'05, pp. 187–195 (2005)
32. Tobii Technology: Tobii Studio 2.2 User Manual (2010)
33. Tobii Technology: Accuracy and Precision Test Report: Tobii T60 Eye Tracker (2011). Date: 2011-07-21, Version: 2.1.1