Visual Data Cleansing of Eye Tracking Data

Christoph Schulz, Michael Burch, and Daniel Weiskopf

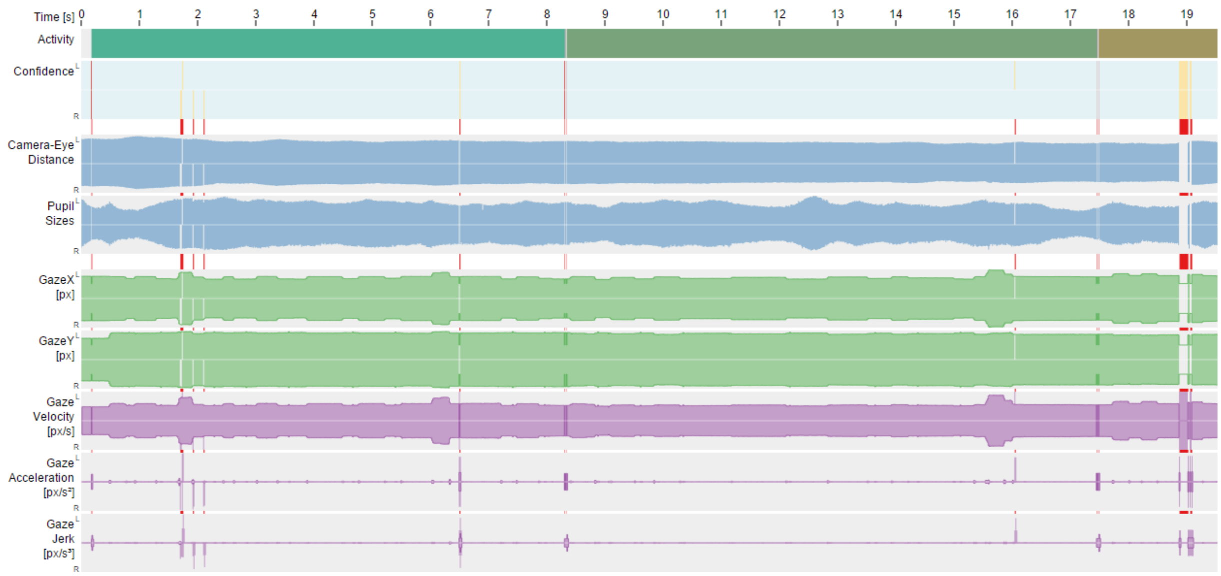

Fig. 1. Time-varying eye tracking data for a sequence of static image stimuli: The recorded data is visualized as stack of time-aligned plots. Differences between the left and right eye can be detected by visually comparing the length of plots mirrored on corresponding horizontal lines. Differences between the individual eyes and the combined eye can be detected by visually comparing the darker line drawn on top of the brighter plot. From top to bottom: time, activity, recognition confidences, normalized camera–eye distance, normalized pupil size, decomposed gaze coordinates, velocity, acceleration, and jerk.

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, since 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 a wide range of eye tracking experiments. The visualization helps researchers get insights into the quality in terms of uncertainty—not only for both eyes in combination but each eye individually. Furthermore, we discuss uncertainty originating from eye tracking, how to reveal it for visualization and illustrate its usefulness using our approach by applying it to eye movement data formerly recorded with a Tobii T60XL stationary eye tracker using a prototypical implementation.

Index Terms—Eye-tracking, low-level data, time-varying data, data cleansing, uncertainty.

1 Introduction

Eye movements recorded during eye tracking studies are typically analyzed and visualized by temporal aggregation like in attention maps [13]. While this allows us to derive hot spots [5] of visual attention, we cannot analyze time-varying patterns. If gaze plots [8] 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 [4], but most of them only take aggregated eye movements into account.

We start earlier than the typical process 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 in Section 3. Our contributions are a discussion of uncertainty in the context of eye tracking in Section 4, a formal cleansing technique for time-series in Section 5, and a visualization technique in Section 6 that reveals disagreement between the left, right, and combined eye.

We demonstrate our visualization technique by applying it to eye tracking data from a previous eye tracking study conducted with a stationary Tobii T60XL eye tracking device in Section 7. As a major outcome, we find differences over time between left, right and combined eye while visually inferring credibility of the recorded data.

To avoid misunderstandings, since work-in-progress results are presented: the described visualizations to locate suspicious data are fully implemented, but emitting of cleansed data is not.

All authors are with the Visualization Research Center of the University of Stuttgart, Allmandring 19, 70569 Stuttgart, Germany. E-Mails: christoph.schulz@visus.uni-stuttgart.de, michael.burch@visus.uni-stuttgart.de, daniel.weiskopf@visus.uni-stuttgart.de.

2 Related Work

Much of the previous work on both eye tracking and data cleansing occurs within isolated domains. Holmqvist et al. [11] already noted that standard 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. To our knowledge, it

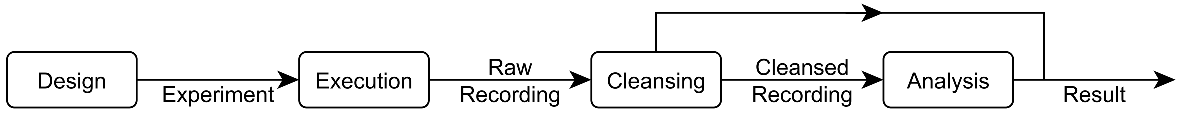

Fig. 2. Reference workflow for eye tracking experiments. Experiments are designed and executed to gain recordings, which can be cleansed and analyzed to obtain a result. The final result is composed from a description of how the cleansing was performed and acquired insights.

seems much more common to enhance study quality by reducing measurement errors introduced by sampling frequency [2] or user movement [7][3], instead of communicating uncertainty present in recorded data. Singh et al. [17] and Al Rahayfe et al. [1] provide reviews of anatomical and technical aspects of eye tracking in general, which are used for discussion later on.

We apply a rationale by Skeels et al. [18] 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. [6].

Rahm et al. [14] 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. [12] 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. [9] 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, like eye tracking.

We apply uncertainty visualization and data cleansing to eye tracking to create a tool for semi-automatic analysis and cleansing of low-level eye tracking data.

3 Experiment Workflow

In advance to discussing uncertainty in eye tracking, we will elaborate on the integration of our tool into a reference experiment workflow, like shown in Figure 2: An eye tracking experiment is designed by a researcher and executed to obtain a recording of time-varying data for each participant. We explicitly refrain from using the term *eye tracking data* or similar on a workflow level, because other data might be recorded as well during execution. Raw recordings may be very hard to compare and analyze, because the order of stimuli may vary by experiment design, recordings may contain recognition errors and originate from multiple eye trackers, and so on.

We believe that proper and comprehensible data cleansing prior to analytics can reduce data quality issues by repairing corrupt data and making heterogeneous data consistent. Furthermore, we propose incorporating a description of how the cleansing was done into the final results, since cleansing has the same potential to hide errors, as fixation filters do, like noted by Holmqvist et al. [11].

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 keep our approach generic. We assume that data is a time-varying series of samples, which allows us to split data into time segments and take a semi-automatic approach to data cleansing. For implementation reasons, we limit ourselves to stationary eye-tracking setups using video stimuli with a duration of less than one hour per recording, even though our concepts should transfer well to any type of time-varying data.

4 Uncertainty in Eye Tracking Data

Our cleansing technique is based on one question: What reveals and separates flawed from trustworthy data, and thus is crucial for decision making during cleansing? This question leads to the topic of uncertainty, which introduces new complexity and data analysis challenges.

4.1 Uncertainty Model

We will adhere to a classification by Skeels et al. [18] to discuss different aspects of uncertainty in the context of eye tracking. Their classification is simple and distinguishes between measurement, completeness, and inference, orthogonally to disagreement and credibility. We apply their classification to a simplified eye tracking pipeline, condensed from related work [1][17][16][19] and depicted in Figure 3, to illustrate sources of uncertainty originating from eye tracking. The process of optical eye tracking starts with photons hitting sensor pixels on a camera chip, aggregated during exposure time to a sequence of images, forming a video. This process introduces measurement uncertainty because of physical properties of pixels such as size and signal-to-noise ratio. Additionally, completeness uncertainty is induced, because of aggregation and sampling. Subsequently, each eye is recognized independently, fused into a combined eye, and synchronized with other data, 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 [20]. 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 their sub-systems have to be considered as block boxes. Unfortunately this means we have to rely on information provided by vendors, which limits our basis for revealing uncertainty and prevents us from finding a common denominator.

Nevertheless, we believe that visualizing uncertainty helps researchers find disagreement in data and estimate a credibility of their recordings. Formally, we try to model and visualize uncertainty as probability density functions (PDF) and nominal data.

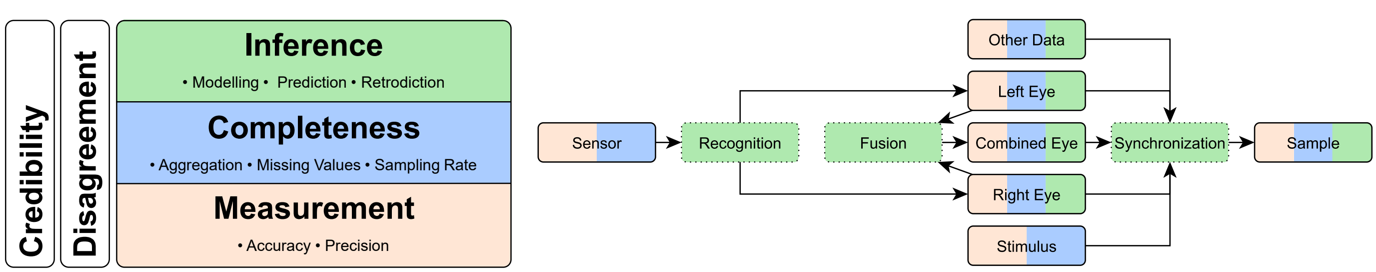

Fig. 3. Reference eye tracker pipeline, decomposed into different layers of uncertainty. Each eye is recognized individually and then fused to a combined eye. Subsequently, all data is synchronized and emitted as sample.

4.2 Gaze Data

When inspecting low-level data, we do not want to redo all work the eye tracker has already done for us, even though inspecting raw sensor data could help assess recognition errors originating from reflections and blinks. Instead, we will focus on uncertainty present in gaze data for the left eye L, right eye R, and combined eye C at time T. Spatial uncertainty can be modeled using PDFs p to describe accuracy and plausibility. Temporal uncertainty can be modeled using rectangular functions i to describe time intervals. Formally, this boils down to a set of tuples:

$$(i(T), p(L), p(R), p(C)) \tag{1}$$

Furthermore, we need a solid definition of space to create useful visualizations. Unfortunately, it is not always possible to determine the actual eye–stimulus transformation, because depth may be approximated or unknown. Most eye trackers make an educated guess by defining the recorded video as stimulus, since the camera–eye distance can be measured quite well. Although this is a clever approximation, it poses problems when dealing with multiple recordings, because finding a common space can be a really 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, like signals S and events E with parameters P_n, that also require cleansing; for instance, the steering angle can be of interest in an automotive context. Temporal uncertainty i and j also apply to signals and events. Temporal uncertainty for samples and events is different, because events are not sampled. Spatial uncertainty of signals can also be modeled as PDF g, which is different from PDF p used for gaze data, since signals most likely can not be checked for plausibility. Formally, this also boils down to a set of tuples:

$$(i(T), g(S)) \tag{2}$$

$$(j(T), E, P_1, ..., P_n) \tag{3}$$

This rather generic definition fits all remaining aspects of a recording, i.e., stimuli changes, latencies, trajectories, keyboard, mouse, and touch input.

5 Cleansing Technique

Our processing model is non-destructive, i.e., all data is interpreted as immutable stream of data and processed by a pipeline composed of functions—this is also called the pipes and filters pattern. A function can perform any non-destructive mapping on a stream. If a stream is pushed or pulled, all affected streams are updated according to their dependencies in the graph. We have chosen this design, because it allows some advanced scenarios such as live monitoring and live usage with other tools. Formally, we want to setup a processing graph $G(f, c)$ composed of functions f and connections c. During cleansing, we want to interact with a time-series domain (t, x) and inspect the resulting codomain (t', x'):

$$f \in G\colon\ (t, x) \rightarrow (t', x') \tag{4}$$

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 differentiating function up to three times.

Interpolating Data 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.

Fixation Filtering is employed to prepare low-level gazes for high-level analysis. Hence, it seems like an obvious choice to incorporate fixation filters into cleansing.

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 pretty tedious task. Therefore, we had to come up with an idea to spare users from small-scale work. Our approach is to split data into time-segments of interest in a semi-automated fashion and let the user validate time segments during cleansing. The main idea is to employ functions to the processing graph that analyze signals, such as recognition confidences or other quality metrics, to emit time segments. Assuming the processing graph was setup correctly, this approach allows going from macroscopic to microscopic cleansing very quickly without missing anything. Creating the processing graph is iterative, hence we distinguish between single and multiple occurrence:

Single-Occurrence segments appear rarely during a cleansing session. As an example of such intervals, the participant might have done something invalidating, which was noted by the supervising researcher. Formally, the domain is tuned manually and the function only applied to one segment. This is a time-local modification to the processing graph and thus allows implicit validation after the modification is applied.

Multi-Occurance segments are either easy to detect or occur more commonly, i.e., the user will likely feel the need for automation. As an example of such intervals, the eye tracker might use magic values to convey error states, which have to be dropped prior to analysis. Formally, the function is applied to all segments and its domain is tuned semi-automatically. This requires the user to (re-)validate all affected segments explicitly.

6 Visualization Technique

Our technique uses a stack of specialized, time-aligned visualizations for each type of data. Time runs from left to right and the stack is sorted in alignment to the specified processing graph to support mental retracing of the data flow. Additionally, we use ColorBrewer palettes [10] to encode information into color.

6.1 Stereo Plots

Fig. 4. A stereo line plot showing the left, right, and combined eye. The right eye chart is flipped below the left eye plot. Differences between the individual eyes and combined eye can be detected by comparing the dark line, representing the combined eye, on top of the individual eye plots.

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 rendered on top of both. Stereo plots manifest as line plots and scarf plots [15].

Line plots are used for ratio-scale data, like shown in Figure 4. A line plot can be normalized to emphasize changes with equal scaling for all eyes.

Scarf plots are used for nominal-scale data such as activity and events, like shown in Figure 5. Activities may be concurrent, hence a scarf plot may be subdivided vertically to indicate concurrency.

Fig. 5. A stereo scarf plot showing activity data over time for the left and right eye. The right eye chart is flipped below the left eye plot.

7 Case Study

The case study aims at showing how to apply the visualization technique, i.e., how to identify typical relationships between visual representations of data that indicate an error.

We have used test data from a previously conducted study with five subjects using a Tobii T60XL and Tobii Studio 2.2.8. The test was conducted in a distraction free room, illuminated with diffuse light. This study is particularly interesting, because the subjects had to match a line to some dots, leading to very fast, comparative eye and head movements. Figure 6 visualizes several time slices from the described data. We start with a description of all representations from top to bottom:

Time is represented as simple ruler and measured in seconds. Scaling was chosen so that one sample is represented by one pixel in width.

Activity events are represented as scarf plot. Long bars depict visibility of image stimuli. Short bars (at the end of stimuli) depict left mouse clicks by the participant.

Confidence is represented as stereo scarf plot for each eye individually. Combined eye confidence is not emitted by the eye tracker. Light blue depicts “all fine”, other colors depict “error”.

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, according to Tobii, an estimate of the true pupil’s size. Both plots are normalized to emphasize changes, hence no unit in the legend.

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 & Acceleration & Jerk are derived from gaze coordinates through differentiation and represented as stereo line plot. Again, the combined eye is depicted using a darkened line.

Next, we will visually inspect all slices from Ⓐ to Ⓓ and draft possible steps for cleansing. The red lines between the plots highlight samples for inspection.

Slice Ⓐ shows minor recognition jitter for both eyes in the confidence plot between 1 s and 2 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 eyes. 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 in an analysis, it might be a good idea to reconstruct the missing individual eye data. To stay within our cleansing model this would be a *single-occurance* function that interpolates the missing values, assuming a simple mean value from the combined eye as input.

Slice Ⓑ reveals more fusion heuristics used by the eye tracker, as both eyes were missing around 19 s and the combined eye is still emitted. 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. This could be a problem if analysis is prone to movement, acceleration, or jerk jitter. In our cleansing model, fixing this could be done by using a *single-occurance* function that interpolates smoothly to fill in missing data if both eyes are not recognized. Furthermore, magic values can be observed during image stimulus switching by correlating activity with confidence. This is documented behavior and cleansing can be done using a *multi-occurance* find-and-drop function set to auto-validation, i.e., requiring no further input by the user.

Slice Ⓒ shows that not all small errors are successfully repaired. Around 28 s the recognition of the left eye fails, shortly after the right eye fails. This seems to bail out error correction for a small amount of time, hardening our suspicion against the combined eye data.

Slice Ⓓ shows the same issue as Ⓑ and Ⓒ. We have encountered the same error three times (arbitrary), which is why we would like to apply our previously specified *single-occurance* functions to all time segments to removing all false eye movement from our data. To do so, we upgrade our previously applied functions to be *multi-occurance*. If previously cleansed data changes during this process, this will cause re-validation.

Another interesting observation across all slices is, that the participant’s pupil sizes seem to be unsynchronized. Unless this is caused by the eye tracker, it might be worth investigating from a medical perspective.

Once, we have cleansed and validated all remaining data, we propose a cleansing report. This report would describe the amount of missing and modified data, e.g., as percentages per recording and time segment, which could be easily incorporated into study results.

8 Conclusion and Future Work

We discussed uncertainty 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 from our initial prototype suggest that more interaction techniques are required to explore eye tracking data during cleansing.

We want to extend our approach to measurement uncertainty and fixation filters or rather their visual response. We believe this could be useful for correlation with areas of interest and filter tuning. Furthermore, we want to broaden the scope to heterogeneous data from multiple participants.

Acknowledgments

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

References

[1] A. Al-Rahayfeh and M. Faezipour. Eye tracking and head movement detection: A state-of-art survey. *Translational Engineering in Health and Medicine, IEEE Journal of*, 1:2100212–2100212, 2013.

[2] R. Andersson, M. Nyström, and K. Holmqvist. Sampling frequency and eye-tracking measures: how speed affects durations, latencies, and more. *Journal of Eye Movement Research*, 3(3):1–12, 2010.

[3] M. Barz, A. Bulling, and F. Daiber. Computational modelling and prediction of gaze estimation error for head-mounted eye trackers, 2015.

[4] T. Blascheck, K. Kurzhals, M. Raschke, M. Burch, D. Weiskopf, and T. Ertl. State-of-the-art of visualization for eye tracking data. In *EuroVis STAR*, pages 63–82, 2014.

[5] A. Bojko. Informative or misleading? Heatmaps deconstructed. In J. Jacko, editor, *Human-Computer Interaction. New Trends*, volume 5610 of *Lecture Notes in Computer Science*, pages 30–39. Springer Berlin Heidelberg, 2009.

[6] K. Brodlie, R. A. Osorio, and A. Lopes. A review of uncertainty in data visualization. In *Expanding the Frontiers of Visual Analytics and Visualization*, pages 81–109. Springer, 2012.

[7] J. J. Cerrolaza, A. Villanueva, M. Villanueva, and R. Cabeza. Error characterization and compensation in eye tracking systems. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA ’12, pages 205–208, New York, NY, USA, 2012. ACM.

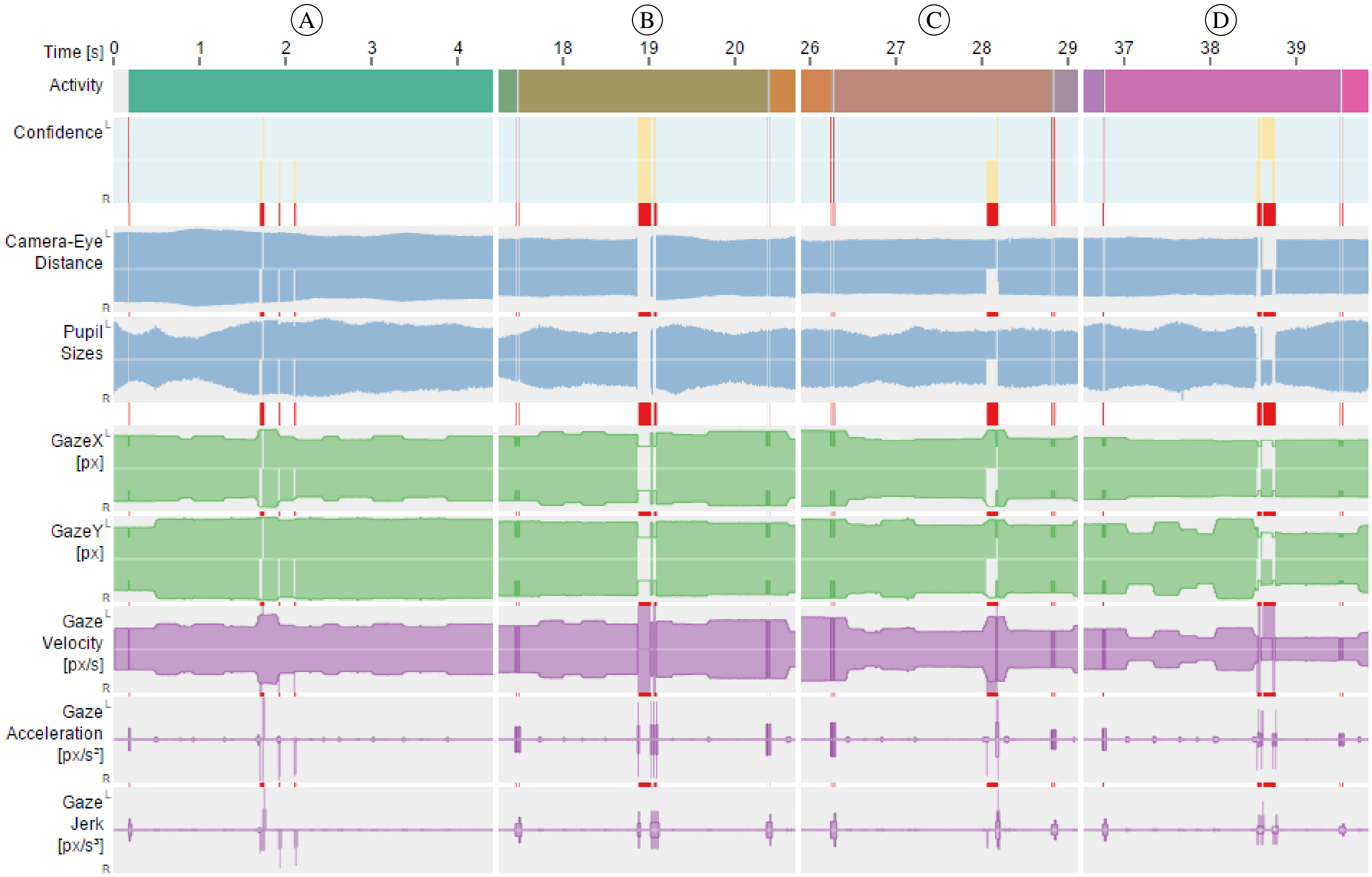

Fig. 6. Four slices from our case study. From top to down: time, confidence, camera–eye distance, pupil size, decomposed gaze coordinates, gaze velocity, acceleration and jerk.

[8] A. Çöltekin, S. Fabrikant, and M. Lacayo. 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.

[9] T. Gschwandtner, W. Aigner, S. Miksch, J. Gärtner, S. Kriglstein, M. Pohl, and N. Suchy. 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, pages 18:1–18:8, New York, NY, USA, 2014. ACM.

[10] M. Harrower and C. A. Brewer. Colorbrewer. org: an online tool for selecting colour schemes for maps. *The Cartographic Journal*, 40(1):27–37, 2003.

[11] K. Holmqvist, M. Nyström, and F. Mulvey. 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, pages 45–52, New York, NY, USA, 2012. ACM.

[12] S. Kandel, A. Paepcke, J. Hellerstein, and J. Heer. Wrangler: Interactive visual specification of data transformation scripts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3363–3372. ACM, 2011.

[13] J. F. Mackworth and N. H. Mackworth. Eye fixations recorded on changing visual scenes by the television eye-marker. *Journal of the Optical Society of America*, 48(7):439–445, Jul 1958.

[14] E. Rahm and H. H. Do. Data cleaning: Problems and current approaches. *IEEE Data Engineering Bulletin*, 23(4):3–13, 2000.

[15] D. C. Richardson and R. Dale. Looking to understand: The coupling between speakers' and listeners' eye movements and its relationship to discourse comprehension. *Cognitive Science*, 29(6):1045–1060, 2005.

[16] SensoMotoric Instruments GmbH. *BeGaze 2.4 Manual*, February 2010.

[17] H. Singh and J. Singh. Human eye tracking and related issues: A review. *International Journal of Scientific and Research Publications*, 2:1–9, 2012.

[18] M. Skeels, B. Lee, G. Smith, and G. G. Robertson. Revealing uncertainty for information visualization. *Information Visualization*, 9(1):70–81, 2010.

[19] Tobii Technology. *Tobii Studio 2.2 User Manual*, 2010.

[20] Tobii Technology. Accuracy and precision test report. Technical report, 2011.