Visual Analytics for Mobile Eye Tracking

Kuno Kurzhals, Marcel Hlawatsch, Christof Seeger, Daniel Weiskopf, *Member*, *IEEE Computer Society*

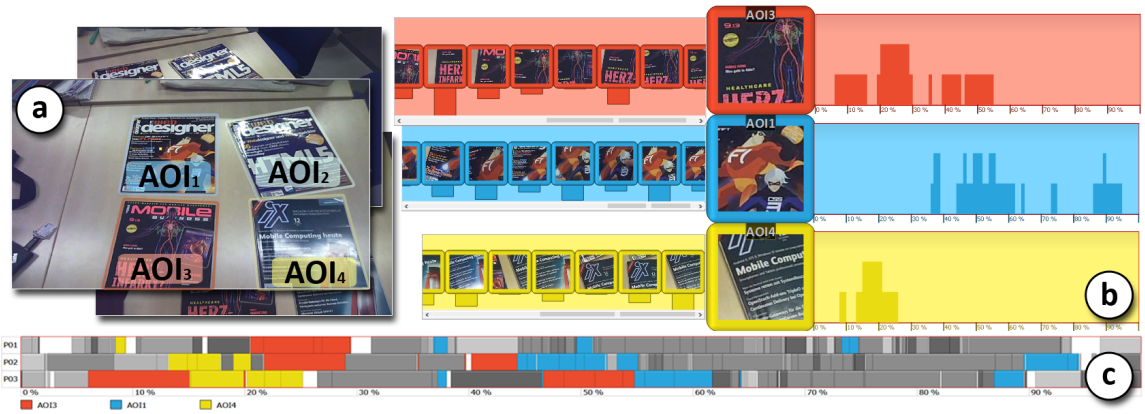


Fig. 1: We provide an interactive analysis approach for mobile eye tracking by visualizing ⓐ gaze data from different videos in a ⓑ cluster view that depicts aggregated information about individual areas of interest. An additional ⓒ scarf plot provides a temporal overview of when annotated areas were investigated by the participants.

Abstract—The analysis of eye tracking data often requires the annotation of areas of interest (AOIs) to derive semantic interpretations of human viewing behavior during experiments. This annotation is typically the most time-consuming step of the analysis process. Especially for data from wearable eye tracking glasses, every independently recorded video has to be annotated individually and corresponding AOIs between videos have to be identified. We provide a novel visual analytics approach to ease this annotation process by image-based, automatic clustering of eye tracking data integrated in an interactive labeling and analysis system. The annotation and analysis are tightly coupled by multiple linked views that allow for a direct interpretation of the labeled data in the context of the recorded video stimuli. The components of our analytics environment were developed with a user-centered design approach in close cooperation with an eye tracking expert. We demonstrate our approach with eye tracking data from a real experiment and compare it to an analysis of the data by manual annotation of dynamic AOIs. Furthermore, we conducted an expert user study with 6 external eye tracking researchers to collect feedback and identify analysis strategies they used while working with our application.

Index Terms—Eye tracking, visual analytics, video visualization

1 Introduction

Eye tracking can be applied in numerous scenarios as a means of measuring visual attention and interpreting visual solution strategies [10]. Areas of interest (AOIs) play an important role because they provide semantic information about investigated regions or objects of potential interest. Measured gaze information can be assigned to the AOIs, serving as the basis for numerous statistical [15] and visual [5] evaluation approaches. To this point, most software suites of the main eye tracking hardware vendors provide the possibility to define bounding shapes as AOIs on the visual stimulus. For static stimuli such as pictures, defining AOIs is much easier to perform than for videos coming from mobile eye tracking glasses. In this dynamic case, AOIs can move, change their size and shape, and even disappear and reappear. Performing manual annotations with bounding shapes is a time-consuming processing step that is prone to annotation inconsistencies, due to the precision issues of the defined bounding shapes.

- *Kuno Kurzhals, Marcel Hlawatsch, Daniel Weiskopf are with the University of Stuttgart. E-mail: firstname.name@visus.uni-stuttgart.de*
- *Christof Seeger is with Stuttgart Media University. E-mail: seeger@hdm-stuttgart.de*

Manuscript received 31 Mar. 2016; accepted 1 Aug. 2016. Date of publication 15 Aug. 2016; date of current version 23 Oct. 2016.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TVCG.2016.2598695

Although the advances in computer vision allow detecting and recognizing specific objects in recorded video material, the so-called semantic gap, "describing the lack of coincidence between the information that can be extracted from the visual data and the interpretation that the same data have for a user in a given situation [30]", is a limitation that still exists to date. Although it is possible to recognize specific AOIs and assign the correct labels to gaze data, all these computer-vision techniques require a trained algorithm that is typically highly optimized for the tested situation. Applying these algorithms to arbitrary eye tracking experiments would require new adjustments and training for any new situation. Without semantic interpretation, the algorithm might be able to detect that there is something interesting in the video data, but providing an answer to the question "what" is typically much more difficult.

With our visual analytics approach, we simplify this complex process by reducing the problem to an image-sorting task supported by automatic image analysis. We follow a concept that involves the analyst in the annotation process by providing both information about the stimulus and the recorded gaze data as small thumbnails (Figure 1). To ease the labeling of thumbnails, we perform unsupervised clustering of the data. The resulting clusters can be explored with different strategies to identify and label AOI-relevant clusters. To support the search for misclassified elements, different image queries can be applied to retrieve the missing thumbnails and assign them to the correct label. Our approach does not rely on the definition of bounding shapes, where the defined shapes might be very different between annotators. Reasons for a low inter-annotator agreement [2] can be investigated easily with our technique by looking at all thumbnails misclassified by the annotators.

In the following sections, we will discuss how our approach relates to other work in this field, what the visualization requirements were, and how we approached them. We evaluated our approach in two ways: we compare the labeled results from our approach with the results obtained by our collaboration partner, for the same real-world dataset. Additionally, we conducted an expert user study at an eye tracking conference to collect feedback about the usability of our visual analytics system and to identify the applied strategies during the use of our application prototype.

2 Related Work

The annotation of AOIs can be performed either by annotating the stimulus content directly, or by labeling the recorded gaze data.

For direct video annotation, manual and automatic approaches exist to extract objects as AOIs. In the best cases, semi-automatic tracking [3, 29] or automatic approaches with markers [25] and without markers [7, 31] facilitate the detection of AOIs. Tracking can improve the annotation speed, but initial definitions and corrections of bounding shapes are still required. Marker-based approaches restrict the possibilities of experiment designs to scenarios where few, specific AOIs can be detected by markers. Additionally, the markers can be a visual distraction in some cases.

Fully automatic identification of AOIs without markers requires an algorithm to detect and recognize the corresponding objects. This is a common problem in computer vision that can usually be solved for specific scenarios (e.g., mobile text recognition [17]), but typically, a training phase with all involved AOIs is required. This prerequisite impairs the application of an automatic approach to solve the annotation issue for arbitrary experiments. We perform image comparisons combined with unsupervised clustering to support the analyst in the labeling process. Our approach requires no initial training phase and can be applied to eye tracking experiments in general.

Labeling the gaze data itself often provides more accurate information about AOIs, since gaze points that were not in an AOI but close to it, for example due to calibration issues, can be identified and corrected. Therefore, each measured gaze point, or in the aggregated case each fixation, has to be investigated in the video to assign the correct label. This annotation approach is in most cases far more time-consuming than the definition of bounding shapes. For example, Tsang et al. [32] depicted fixations labeled this way also by thumbnails. As the authors mentioned: "This process constitutes a significant amount of time and effort if the number of fixations is large [32]". Netzel et al. [24] reported an average annotation speed of 5 fixations per minute by a similar approach, leading to 140 hours spent on about 40.000 fixations for an experiment. There is some work that improves the annotation step by semi-automatic algorithms. Pontillo et al. [26] presented an image-based approach to label fixations by showing images of fixated regions to the analyst for a semi-automatic classification of fixated areas. However, the authors applied this approach only to assign fixations to labels, further analysis with statistical or visualization techniques is still required for this annotated data. Also, their approach required a step-wise labeling of the data, while we allow for an automatic clustering of the images in a pre-processing step, which reduces the number of images to investigate.

Kurzhals et al. [19] depicted measured gaze points by thumbnails from the investigated stimulus regions on separate timelines for each participant. They supported fixation queries based on image similarities on demand. However, their work was restricted to static and constrained dynamic stimuli where the visual stimulus was identical for each participant and only synchronization between the recorded data was required for comparison. Mobile eye tracking, as in our case, is far less constrained, allowing the participants to move freely during an experiment, resulting in individual videos for each participant that are more difficult to compare, due to individual video lengths and perspectives on involved AOIs. Nevertheless, mobile eye tracking becomes more and more popular, allowing "in-the-wild" studies that are not possible with restricting experimental settings. Therefore, the development of more efficient analysis methods for these challenging datasets is an important research field.

Chinchor et al. [8] proposed denoting the combination of multimedia analysis and visual analytics as "multimedia analytics", which would also apply to our approach. There are numerous methods to depict large collections of video data, e.g., Luo et al. [22] analyzed and visualized news video collections according to an interestingness measurement. Borgo et al. [6] provided a survey of the different approaches for video visualization. The cluster editor view of our approach is similar to storyboard visualizations that depict keyframes of videos in a grid (e.g., the work by Bailer and Thallinger [1], Furini et al. [12]). Fu et al. [11] used a similar concept to visualize multiview videos which show the same scene from different views. In our technique, images can be assigned to AOI labels by dragging-and-dropping on their representative pictograms. This approach is similar to MediaTable by Rooij et al. [9, 28]. The authors used a bucket-based workflow to categorize videos. Buckets represented media categories and videos were displayed by representative images. However, their application was developed only for video content without any eye tracking information. We adopt the principles of their workflow to provide an efficient means of labeling AOIs with pre-processed data.

For the visualization of the annotated eye tracking data, we rely on techniques that are most common in the practice of eye tracking. Blascheck et al. [5] provided an overview of state-of-the-art visualization techniques for eye tracking data. From these, we choose two techniques that fit the requirements of our analysis tasks (see Section 3): histograms of visual attention on the AOIs and scarf plots [27].

Our main contribution is a new visual analytics approach that allows the efficient comparison of data from multiple videos acquired during experiments with mobile eye tracking. By including unsupervised clustering techniques in the pre-processing and interactive image queries in the labeling step of the analysis process, we achieve annotation results comparable to current state-of-the-art techniques, but with far less human effort and a more efficient annotation process.

3 Domain-Specific Analysis Process

In order to design a visual analytics approach that facilitates the analysis of mobile eye tracking data, we first had to identify the requirements the technique should fulfill. Based on this, we formulated how the analysis process has to be changed in comparison to a traditional analysis of mobile eye tracking data from multiple participants with dynamic AOIs. Although we designed this approach with a specific application scenario in mind, the general analysis questions we derived apply to a multitude of possible mobile eye tracking experiments.

3.1 Domain Problem Characterization and Design Process

The development of our technique was accompanied by discussions with our collaboration partner (a co-author of this paper). Our partner is an experienced eye tracking researcher (8 years) at the Stuttgart Media University in the field of print media. Following the principles of user-centered design [23], we aimed at the domain problem characterization, and identified the requirements of our collaboration partner. In an iterative process, we discussed, adjusted, and improved the design of our approach. For the domain problem characterization, we identified the following points as the main analysis questions to be answered:

- **Q_1:** What was the distribution of attention between AOIs?
- **Q_2:** When was a specific AOI investigated for the first time?
- **Q_3:** In which order did the participants look at the AOIs?

These are three basic questions for eye tracking analysis tasks, allowing us to apply established visualization techniques and descriptive statistics to present the extracted information in an appropriate way, familiar to eye tracking researchers. To answer the first question, we provide attention histograms, showing the average gaze duration of all participants in relative time. This representation is consistent to the one used by our collaboration partner. The inclusion of additional metrics would be possible to address further research questions. For the other questions, we included a scarf plot for all participants,

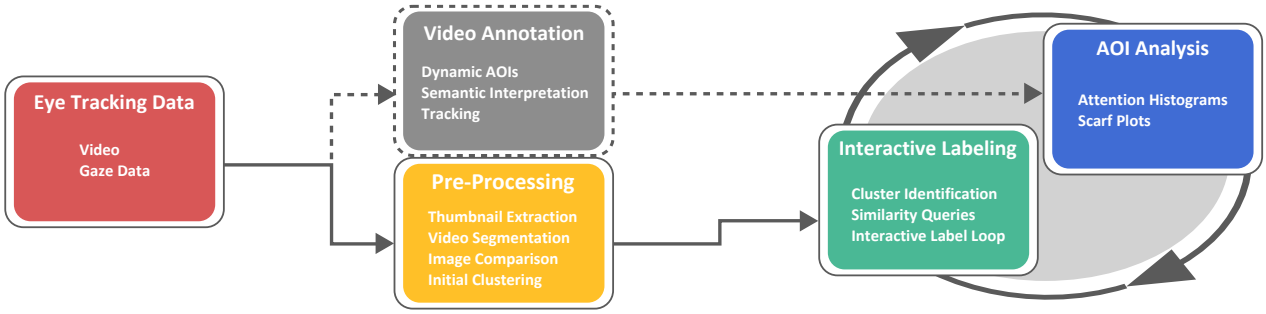


Fig. 2: Analysis process for mobile eye tracking data (red): AOIs can be defined by annotating the video (gray) or the gaze data. We facilitate the annotation by automatic pre-processing of gaze data (yellow) and reduce the annotation to an interactive image labeling task (green). For the analysis of mapped gaze data on AOIs (blue), common visualization techniques such as histograms and scarf plots can be applied.

which consists basically of color-coded timelines of attended AOIs. This technique is a common approach to represent recorded scanpaths of participants with information about visited AOIs [18, 27]. Choosing suitable visualization techniques for interpreting the labeled data was a minor issue during the design process of our visual analytics approach because established methods have been identified to be appropriate.

3.2 Analysis Process

With the domain problem characterization and requirements analysis, we also looked into the typical workflow associated with mobile eye tracking studies. Here, we identified AOI labeling as the by-far most time-consuming step. Therefore, the main focus of our work was on making the labeling process more efficient.

For this purpose, we changed the common process for annotating eye tracking videos with AOIs (see Figure 2). Recorded eye tracking data consists of a recorded video and gaze data mapped to the video. In the traditional annotation process, the video is investigated and dynamic AOIs have to be defined on the video image. Depending on the software employed, this process is supported by tracking algorithms from computer vision that still require the definition of shapes and the correction of tracking results. This procedure has to be repeated for each video (i.e., each participant) recorded in the experiment. Furthermore, consistent labeling of AOIs is critical for the analysis. Gaze points are mapped to the AOIs by automatic hit detection, the analyst is usually not involved in this mapping process. Consequently, errors from imprecise bounding shapes or offsets in the calibration of the eye tracker might be overlooked.

By investigating the image content of fixated regions directly, we provide the analyst the possibility to decide whether a gaze point was on an AOI, or not. However, looking at the image content of each measured gaze sample individually would need much more time than the definition of dynamic AOIs. Therefore, we chose to split the analysis process in two stages (Figure 2): a pre-processing step that can be performed automatically and clusters gaze data based on the investigated stimulus content, and the subsequent analysis of these clusters itself. The analysis can be interpreted as a loop between the interactive labeling of the clusters and the coupled interpretation of the results with the provided visualization techniques: all changes in the labeling can be directly interpreted by the other visualizations. Then, the next clusters can be investigated based on the insight derived from the visualizations. In Section 5, we will describe in detail how our analytics environment supports this interactive labeling and analysis process. Beforehand, we will discuss the pre-processing of the data.

4 Pre-Processing

As illustrated in Figure 2, eye tracking data has to be recorded and pre-processed before the interactive labeling step. For our examples, the SensoMotoric Instrument (SMI) head-mounted Eye Tracking Glasses 2.0 were used. The device records a video from a field-of-view camera and exports gaze coordinates mapped to the coordinate system of this video. We used the fixation filter SMI provides to label raw gaze points as either belonging to a fixation or not. However, our visual analytics

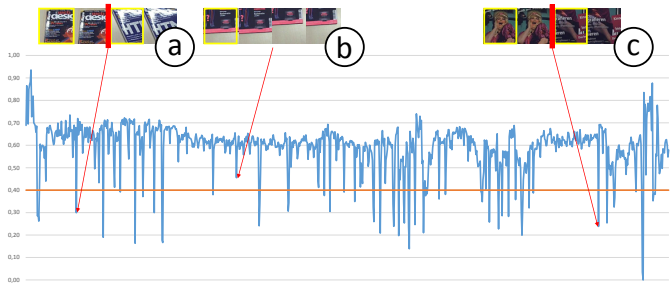


Fig. 3: Segmentation of a thumbnail sequence: Changes in the image sequence lead to similarity values below the threshold (ⓐ, ⓒ) and start a new segment. Smaller changes due to short saccades or head movement are aggregated in the same segment ⓑ. The first element of a segment is chosen as representative (yellow border).

method does not make use of any specific characteristics of the SMI glasses. Therefore, it works with any eye tracking data that provides gaze coordinates of fixations and a video of the stimulus. All image processing steps were performed with OpenCV 3.0[1].

Thumbnail Extraction: Each gaze point provides an x- and y-coordinate mapped to the corresponding video recorded by the scene camera of the eye tracking glasses. Around the gaze position, a thumbnail can be cut out of the video image, representing the currently watched region. In accordance with Kurzhals et al. [20], we create thumbnails with a size of 100×100 pixels to cover the approximated foveated region at an average viewing distance of 65 cm. In general, an increased thumbnail size is advantageous for the detection of image features, but impairs the interpretation of what was investigated by the participant during the experiment.

Video Segmentation: This step describes the temporal segmentation of the video. We first reduce the number of relevant images by taking advantage of the temporal coherence of the underlying video and gaze data. Fixations on a specific area typically result in a sequence of images that are similar. Therefore, we perform a comparison of thumbnails from subsequent video frames and aggregate the images until they drop below a similarity threshold. Depending on the applied similarity measure, the threshold can be adjusted to achieve longer or shorter segments. For our applied measure (see next paragraph), a threshold of 0.4 still provided good segmentation results without aggregating different stimulus regions (Figure 3). We refer to this aggregated sequence of thumbnails as *segment* in the following. A segment is represented by the first thumbnail of the sequence. We stop the aggregation for a segment when more than 2 frames are missing in the gaze data between consecutive thumbnails. This can happen when the eyes are not recognized for a short timespan by the eye tracking device. Depending on the similarity threshold and how much the eyes move during the experiment, this segmentation step can reduce the number

[1] OpenCV: *http://opencv.org*, last checked 2016-06-27

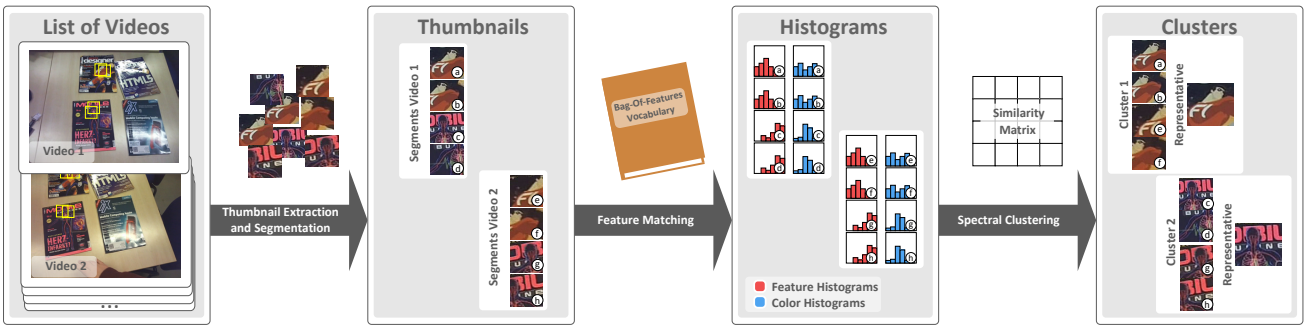


Fig. 4: Overview of the video segmentation and image clustering process: Thumbnails are extracted from all videos and temporally aggregated. The segment representatives are compared using a bag-of-features approach. Clustering is then performed on the resulting similarity matrix.

of images for the subsequent clustering step to approximately 10% of the original thumbnails, removing redundant images from fixations on the same regions. Other experiments that involve smooth pursuit eye movements should aim for smaller segments, since the motion of the underlying stimulus content might be hard to interpret when it is represented only by a single image [19].

Image Comparison and Clustering: As described above, we compare thumbnail similarity for two reasons: segmentation of the image sequences and clustering of the remaining segments. From the numerous possibilities to compare images, we combine two approaches that require only a few parameters and can be applied to arbitrary image sequences (Figure 4). The first similarity value is calculated from extracted SIFT features [21] using a bag-of-features approach [16]. We extract the features from each thumbnail and create a feature vocabulary, using k-means clustering. For our tested examples, a set of maximal 200 features per image and a vocabulary size of $k = 500$ led to good results for the segmentation. In general, the size of the vocabulary depends on the number of regions to differentiate. However, it has to be considered that a large vocabulary size might cause overfitting.

With the vocabulary, feature histograms can be derived for every image. At this point, this approach is typically applied to train a classifier for a specific image category. Since this would already require gold labels of the AOIs, we pass this step and calculate the similarity between the feature histograms of two images, using the inverted Bhattacharyya distance [4], which provides us with a normalized similarity value. Since the extraction of SIFT features depends on the quality of the investigated images, some of the analyzed thumbnails provide either few or no good features to extract. To compensate for that, we include a second image similarity measure, based on a color histogram comparison of the hue and saturation values of the images. We neglect luminance to make the comparison more robust against changes in illumination [33]. Both similarity values, feature-based and histogram-based, are aggregated equally. In the case that the feature recognition of an image fails due to a low number of recognized features, only the color histogram value is applied.

Unsupervised clustering of the thumbnails is performed on their similarity matrix, using self-tuning spectral clustering [34] that only needs a maximum number of clusters as parameter. For the preprocessing, this should be at least the number of required AOIs. Since irrelevant regions and large AOIs will lead to sub-clusters, the maximum number of clusters should be adjusted accordingly. For our experiments, 50 clusters separated the images sufficiently to initialize the labeling process. For each cluster, a cluster representative is determined by calculating the thumbnail that is most similar to the other thumbnails in a cluster.

Note that the applied similarity measures and clustering method were a choice of current state-of-the-art methods that work in general for most experimental settings. The similarity measures could be replaced by others that apply better to the specific requirements of the recorded data. Therefore, we refer to Smeulders et al. [30] for an overview of other possible image retrieval techniques. As a result of the pre-processing, we acquire a list of clustered thumbnails and the calculated similarity matrix.

5 Analytics Environment

We designed our analytics environment in a way that the analysis process is effectively supported (see Figure 2) and that the questions ($\mathbf{Q_1}$–$\mathbf{Q_3}$) of our collaboration partner can be efficiently answered. For this, we integrated a number of components (see Figure 5) which allow an effective analysis of the data and provide important information related to these questions. The analysis is performed on the clustering results from the pre-processing step (Section 4). So far, the clustered data does not contain any semantic interpretation of AOIs and misclassified segments can appear in the clusters. Therefore, our analytics environment supports an intuitive labeling process, the detection of falsely clustered elements, and the modification of clusters.

The main view of our implementation allows performing all AOI analysis tasks and parts of the interactive labeling (Figure 2). To further improve the interactive labeling task, additional views are provided: the cluster editor to inspect, modify, create, or delete clusters. The video player to investigate the video stimuli and gaze behavior of individual participants and to search for segments by defining AOIs directly on the video. With these different views, it is possible to apply different strategies to label and analyze the data. In our use case (Section 6) and the description of our expert user study (Section 7), we will discuss how to approach a typical analysis scenario.

5.1 Main View

Cluster view: The central component of our main view is the cluster view in the center (Figure 5 ⓐ). It lists the clustered segments of the eye tracking data (see Section 4). Each cluster is depicted by a cluster representative computed during the clustering process (Figure 4), which is shown enlarged on a vertical axis. To the left of each cluster representative, all segments inside the cluster are shown (Figure 5 ⓐ$_1$). They are sorted according to their similarity. The initial view shows the thumbnails with lowest similarity on the left. In this way, the user gets an impression of the quality of the cluster and mismatching thumbnails might be found directly without additional exploration. To the right, an attention histogram shows the occurrence of the cluster accumulated over all participants (Figure 5 ⓐ$_2$), i.e., the histogram value is determined by the number of participants that looked at the respective segments contained in the cluster. Since the recorded videos have different durations, our timeline representations are calculated in relative time in order to make the data comparable between participants. To provide a quick overview of the potentially most important parts of the data, the cluster list is sorted according to the accumulated gaze duration of the clusters.

The cluster view serves as a starting point of the analysis by allowing the investigation and labeling of the pre-computed clusters; label colors and names can be assigned to the clusters in this view. It is also possible to modify the clusters in this view by dragging individual elements to other clusters. However, if the clustering quality is not satisfactory and larger modifications are required, this is better performed in the cluster editor (Section 5.2), which can be opened by simply double-clicking on cluster representatives.

Coupled with the interactive labeling process, the components for the AOI analysis are updated accordingly. The attention histograms ⓐ$_2$

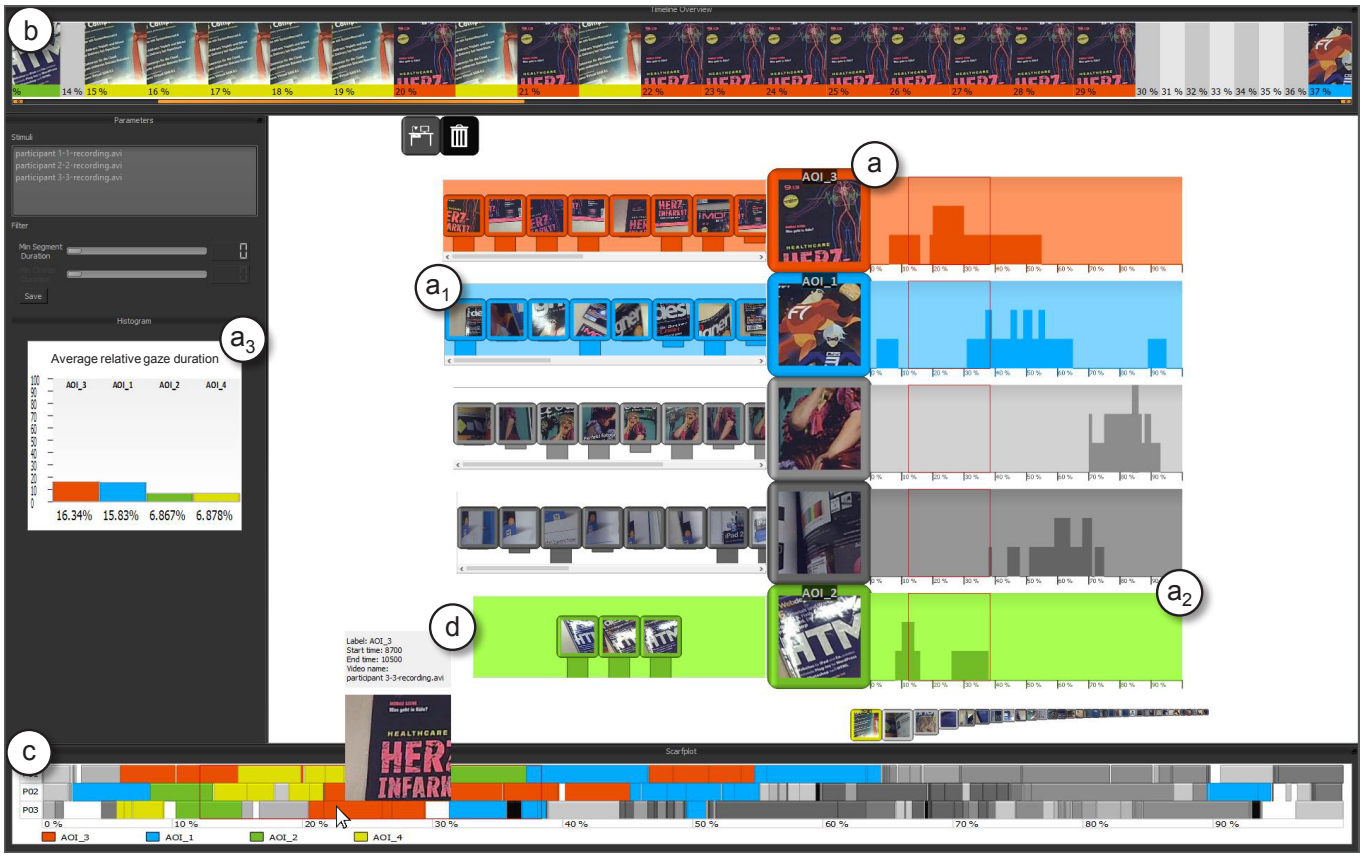


Fig. 5: Overview: Our implementation consists of a main view with four different elements: (a) the cluster view lists all clusters in the data sorted by their accumulated duration. (a_1) To the left of each cluster representative, the cluster elements are shown. (a_2) To the right, attention histograms for the clusters are shown. (a_3) The total attention of the labeled clusters is shown in the histogram on the left. (b) The timeline overview at the top shows the clusters that are viewed by the majority of the participants. (c) The scarf plot at the bottom displays for each participant over time which cluster was investigated. (d) A tooltip shows additional information when hovering over scarf plot segments.

can help to partially answer the question when an AOI was visited the first time ($\mathbf{Q_2}$). However, this visualization is better suited to find out if there are timespans during the experiment when many participants looked at the same AOI (e.g., reading the caption of an article only in the beginning). An even more aggregated version of the histograms is shown on the left (Figure 5 (a_3)). It shows the average relative gaze duration of all labeled clusters. This allows seeing directly which AOIs received most attention by the participants ($\mathbf{Q_1}$). In general, additional descriptive statistics could be integrated in this view. So far, we included only the gaze duration for our comparison in the use case (Section 6). However, these histograms do not allow solving all of our analysis questions (Section 3) efficiently. For this, we provide two additional views integrated into our implementation.

Timeline overview: The timeline overview is displayed on top of our implementation (Figure 5 (b)). This view shows a cluster representative on the timeline if the number of participants looking at the cluster segments is over a user-defined threshold. In this way, the timeline overview presents a summary of what the majority of participants looked at, i.e., it can be seen as an accumulated attention histogram showing only the clusters with high attention over time. The part of the timeline that is currently shown is also marked in the other views with a red box so the user can analyze how this summary correlates with the cluster histogram and the scarf plot on the bottom. Furthermore, label colors are also shown to ease the identification of clusters and the temporal position is displayed as percentage of the video length. With the timeline overview, answers for questions $\mathbf{Q_2}$ and $\mathbf{Q_3}$ (Section 3) in terms of an "average scanpath" can be easily found by looking for the first appearance of a cluster or by investigating the order of the clusters on the timeline.

While vertically stacking the cluster representatives for the same time would ease the interpretation, we decided to stack them horizontally to keep the view compact and avoid vertical scrolling. However, since the feedback from our user study (Section 7) showed that users had problems with this view, we want to improve this component in the future with a better design. Furthermore, the timeline provides only information accumulated over all participants; a detailed analysis of individual participants is not possible with it. For such an analysis, a scarf plot was integrated into our analytics environment.

Scarf plot: The scarf plot at the bottom shows the data of individual participants (Figure 5 (c)). For each participant, a timeline is shown with colored blocks representing the different extracted segments. The length of a block corresponds to the duration of a segment, i.e., how long a participant looked at a specific region. Initially, different shades of gray are automatically assigned to the clusters. These are replaced by the label color when the user assigns a label to the cluster. By double-clicking on a specific segment in the scarf plot, the cluster editor opens the corresponding cluster the segment belongs to. The analyst can then label the cluster and its segments will appear in the color of this label in the scarf plot. Gray segments between segments of the same color can be an indicator of faults in the clustering results. The analyst can investigate these segments and label the respective clusters, coloring the scarf plot iteratively with every new cluster. Hence, the loop between the labeling and the direct analysis of the labeled data can be repeated until all relevant information is found or the complete dataset is labeled.

With this scarf plot, we can easily see when and how long different participants looked at specific clusters, providing us with detailed information to answer the questions $\mathbf{Q_2}$ and $\mathbf{Q_3}$. When hovering with the

Fig. 6: Cluster editor. The cluster editor allows editing, merging, and deleting individual clusters. It shows a list of all clusters and in the center the elements of the selected clusters. By dragging and dropping cluster elements, they can be assigned to other clusters or deleted. Labeled clusters and their elements are shown with the label color. When hovering an element with the mouse, a tooltip shows the full video frame with the gaze point and thumbnail border marked.

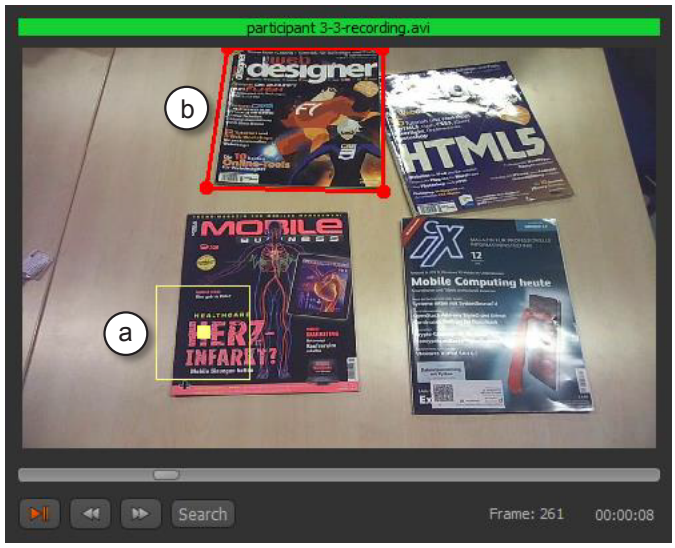


Fig. 7: Video player. With the video player, recordings of individual participants can be viewed. ⓐ Their gaze positions are marked with a yellow square and bounding box. ⓑ A polygonal area can be marked (red) in the video frame to search for all similar looking thumbnails.

mouse over a block of the scarf plot, additional information are provided (Figure 5 ⓓ): an image showing what the participant looked at, the duration of the block in milliseconds, the label, and the name of the corresponding video.

With this view, it is possible to see how the gaze of individual participants moved between different AOIs. Viewing behavior can also be compared between participants to detect outliers or common patterns.

5.2 Cluster Editor

The effectiveness of the analysis with our main view strongly depends on the quality of the clusters, i.e., how well they represent specific AOIs in the stimuli. The results are only meaningful if a cluster contains all relevant elements of an individual AOI. Since the clustering algorithms (see Section 4) are not perfect, we allow manual verification and modification of the clusters. For this, we integrated the cluster editor (Figure 6).

The cluster editor is divided into three parts: a list of labeled clusters at the top, a list of non-labeled clusters on the left, and the segments of a selected cluster in the center. Clusters are selected by clicking on them; their elements are then shown. Unwanted elements of a cluster can be deleted (by dragging the element on the garbage can symbol) or moved to other clusters (by dragging the element on a cluster representative). Dragging an element onto the desk symbol allows collecting different thumbnails of potential interest (e.g., ambiguous thumbnails) for a further inspection later on.

Integrated image search further supports editing the clusters. It is possible to select a thumbnail and let the system search for similar thumbnails in the same cluster, in the other clusters, in all clusters, and in unlabeled clusters. The found thumbnails are then ordered according to the image similarity metric (Section 4), derived directly from the similarity matrix. Image queries can be processed very efficiently by sorting the row of the similarity matrix of the currently searched thumbnail. With this function, it is quite easy to find similar thumbnails in other clusters or perform cluster corrections, e.g., splitting a cluster into two clusters: a thumbnail with the specific content for the new cluster is selected and used for image search. The thumbnails are then ordered according to image similarity allowing an easy rubber band selection to create a new cluster.

In some cases, it is hard to decide from the thumbnails alone if a certain element belongs to a cluster because the surrounding context of the thumbnail is missing. Therefore, the complete video frame is shown when hovering with the mouse over a thumbnail in the cluster editor. In the video frame, the gaze position and the thumbnail bounding box are marked. This eases the interpretation of thumbnails by providing the full context of the stimulus.

5.3 Video Player

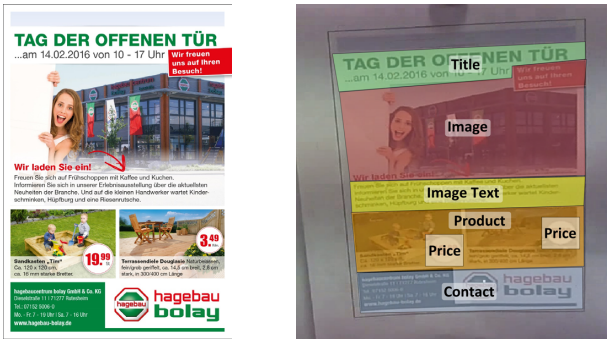
Without knowing the content of the stimuli, it is difficult to understand the data and what the clusters and thumbnails represent. Therefore, it is possible to watch the recordings of individual participants in a video player. The video player (Figure 7) shows the video recorded by the eye tracking hardware together with the respective gaze positions (Figure 7 ⓐ). This supports not only the general understanding of the data by showing the stimulus content, but also allows following the gaze movements of an individual participant. Furthermore, it is possible to select a polygonal area in the video frame (Figure 7 ⓑ) and perform an image search with the selected area. The thumbnails (Section 4) that are most similar to the selected region are then shown in the cluster editor. In this way, it is possible to create clusters by drawing AOIs in the video as in the traditional approach (Section 3).

6 Use Case

To evaluate our method on eye tracking data from a real experiment, we applied our technique to the data from our collaboration partner. The data was recorded from an eye tracking experiment in a hardware store. The experiment was part of a research project at the Stuttgart Media University. The question was, how different designs of printed advertisements affect the perception of viewers. Since the evaluation of the experiment was also performed by our collaboration partner with traditional methods, we can provide a comparison of the annotation time and the distribution of attention on different AOIs.

6.1 Design of the Eye Tracking Experiment

The stimuli were categorized in three sections of intentional dimensions: *Sale*, *Image*, and *Event*. For each of these dimensions, two different design categories were tested with eye tracking in a between-subject design and an additional post-test interview to compare the distribution of attention on the different stimuli. The first category was a positive design according to the intention, the second category was not. The entire experiment took place at the point of sale in a hardware store, where regular customers were faced with one stimulus after they agreed to be a part of the experiment. In total, 90 persons



(a) (b)

Fig. 8: (a) Visual stimulus from the investigated user study. (b) Annotated regions on the stimulus represent the labels of the relevant AOIs.

participated in the experiment. For our comparison, we investigated one of the six stimuli (Figure 8) that is defined as a design with the intention *Event* but also consists of other design objects, like prices and product pictures. 15 participants looked at this stimulus for approximately 20 seconds each, two of them were removed due to calibration issues. For the application of our approach to the complete experiment, the described annotation process can be repeated for the stimuli of the other intentional dimensions.

6.2 Comparison

The traditional analysis procedure of eye tracking data (Figure 2), including the annotation of dynamic AOIs directly in the videos, was performed by a group of four students of our collaboration partner using the SMI software BeGaze. To achieve comparable results between the approaches, we defined the same stimulus regions as AOI labels (Figure 8b) and assigned the extracted segments with our technique accordingly. The labeling process was performed by two of the developers, providing a first impression of how efficient trained users can apply the technique.

Figure 9 shows the resulting annotation times. For dynamic AOIs, each video has to be investigated individually and the consistency of the annotated areas has to be considered. This requires concentrated drawing and correction of polygons over time. In comparison to this traditional approach, we could reduce the annotation process to approximately 17%–30% of this time, depending on the applied annotation strategy. Annotator 1 used direct searches of areas by drawing query regions in the video. Since these searches required time to process, the annotation was slower. Annotator 2 iterated through the clusters, using only the pre-processed similarities between thumbnails to search segments belonging to an AOI, which provided almost instantaneous query results. Since we used a non-optimized algorithm to search for the arbitrary image queries, we argue that Annotator 1 would also have finished earlier if the calculations were more efficient.

In Section 3, we discussed the requirements and research questions our visual analytics approach should be able to answer. First, we have to be able to derive the distribution of attention on different AOIs ($\mathbf{Q_1}$). Since this was also the main question of our collaboration partner, we can compare their results, derived by dynamic AOIs, with our approach. Figure 10 shows the average relative gaze duration on the different AOIs. Note that in our case no ground truth is available to compare to. Hence, we focus on the comparison between the two techniques. In summary, we could achieve congruent results with our approach with differences between 0.4%–6% in comparison to dynamic AOIs. For an average video duration of 20 seconds, the maximum difference between the calculated attention on the AOI *Product* is 1.2 seconds. Between the two annotators who applied our approach, the differences are between 0.1%–2.6%. We noticed that especially fixations in border regions lead to variations in the annotation results. Depending on the size of the drawn AOIs, some fixations might be neglected

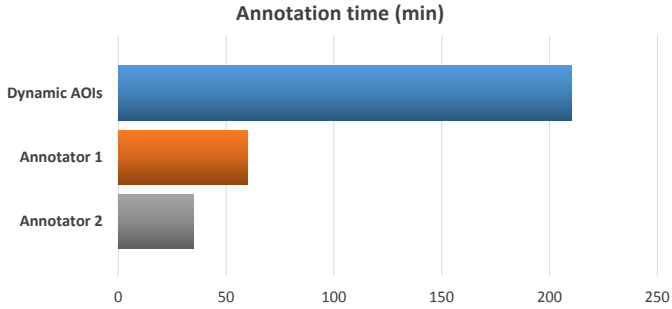


Fig. 9: Annotation times for thirteen videos. The dynamic AOIs were labeled directly in the video by drawing and tracking bounding shapes. Annotator 1 and 2 applied our approach. Their different completion times result from different strategies applied to solve the task.

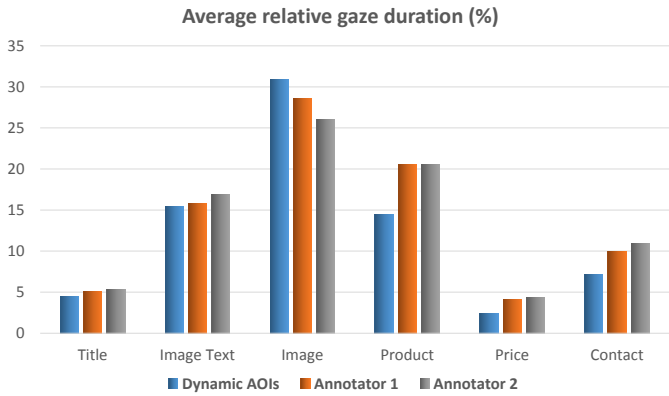


Fig. 10: Comparison of dynamic AOIs with our approach. Differences mainly result from gaze points in boundary regions of the AOIs.

even if an annotating human user might assign it to the corresponding label. With our approach, such difference in the inter-annotator agreement could be solved by displaying the issued thumbnails in the editor view and let the user decide where a segment belongs to.

To answer the other two questions from Section 3, when AOIs were watched ($\mathbf{Q_2}$) and in which order ($\mathbf{Q_3}$), the annotated scarf plot (Figure 11) can be interpreted. For example, we can see which participants focused more on the images (red, orange) and when participants started to read the image text (long yellow segments). The black areas indicate that the segments were moved to the garbage can, since they did not belong to any AOI.

7 Expert User Study

Our use case showed how trained users can apply the technique. To gather qualitative feedback how untrained but eye tracking experienced researchers can work with our technique, we conducted a user study at the *Symposium on Eye Tracking Research and Applications (ETRA 2016)*. Due to temporal restrictions, we used a smaller dataset in this study: the dataset consists of 3 videos with the participant standing in front of four magazines, looking at the covers and picking up one of them to browse through (Figure 7). All three videos have an approximate length of 30 seconds. The four magazine covers are the AOIs, starting with AOI 1 in the upper-left corner and ending with AOI 4 in the lower-right corner (see Figure 1). The data was recorded with a free-viewing task as showcase for the expert study. We asked 6 experts (age 28–40 years) with different degrees of experience in eye tracking (5–10 years) to use our technique to label the four AOIs. Their research fields were psychology, software engineering, virtual reality, and spatial cognition. Additionally, a freelance developer of eye tracking applications participated. The study took about 45 minutes on average, including an introduction and a demonstration of how to use the different components.

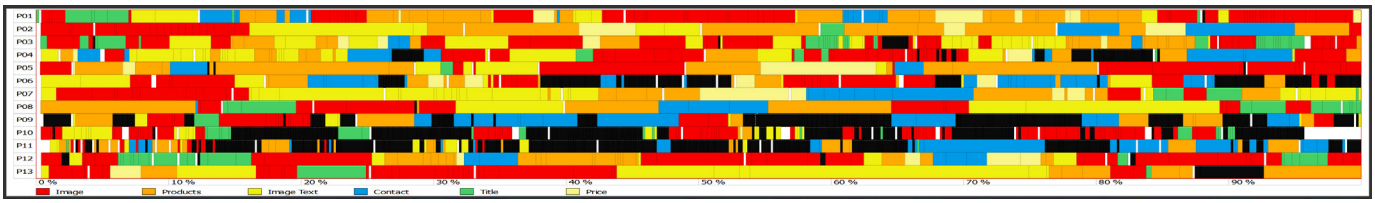


Fig. 11: Scarf plot of 13 participants labeled with our approach. Black areas depict segments that were put to the garbage can due to gaze points outside the AOIs.

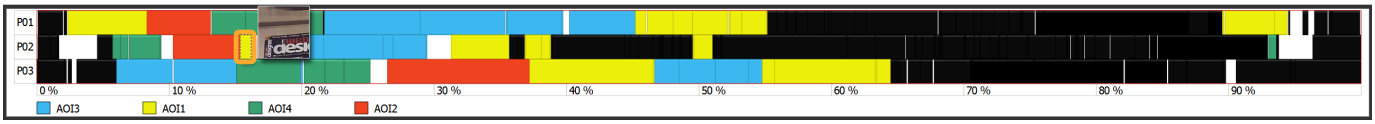


Fig. 12: Scarf plot of the data that was analyzed in the expert user study. A yellow segment (marked orange) was misclassified by two participants.

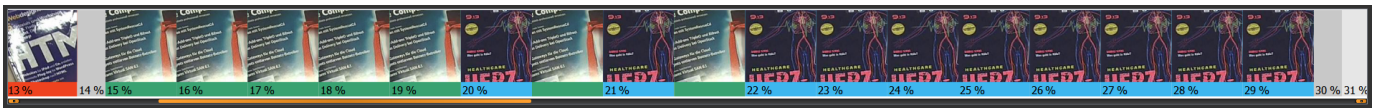


Fig. 13: Timeline overview showing the longest relative time interval when two participants looked at the same AOI.

Each expert was introduced to the software by a two-sided sheet, explaining the main functions and views. As an example, one cluster from the dataset (not relevant for the following task) was labeled and analyzed to show all functionalities. Then, the experts were free to apply all available functions to label the 4 AOIs, in order to answer three questions about the order in which participants looked at the AOIs, the distribution of attention, and the longest common timespan two participants spent on the same AOI.

We asked the experts to start with defining the labels of all four AOIs to proceed with the labeling in parallel, in order to prevent that they are slowed down by a sequential search of individual AOIs. An additional questionnaire was handed out to rate the visualization components and collect qualitative feedback about the approach.

7.1 Results

First, we investigated the results for the three questions the experts answered by interpreting the different visualization views:

1. Which was the order the participants looked at AOI_1 – AOI_4?

   P_1: $AOI_1 \rightarrow AOI_2 \rightarrow AOI_4 \rightarrow AOI_3$

   P_2: $AOI_4 \rightarrow AOI_2 \rightarrow AOI_1 \rightarrow AOI_3$

   P_3: $AOI_3 \rightarrow AOI_4 \rightarrow AOI_2 \rightarrow AOI_1$

   Figure 12 shows the scarf plot of the data. Only the four task-relevant clusters were labeled, the other clusters were removed. For P_3 all experts answered correctly, for P_1 only one answer was not correct. For P_2 two experts gave the wrong answer. In this case, we identified that the small segment from AOI_1 between 10% and 20% (Figure 12) was not labeled correctly.

2. What was the average relative gaze duration on AOI_1 – AOI_4?

   AOI_1 - mean: 15.68%, sd: 0.95%

   AOI_2 - mean: 6.87%, sd: 0%

   AOI_3 - mean: 16.34%, sd: 0%

   AOI_4 - mean: 6.91%, sd: 0.86%

   The distribution of attention shows that the attention on AOI_1 and AOI_3 was higher than on the other two AOIs. For AOI_2 and AOI_3, the standard deviation is zero, since all relevant images were labeled by the experts identically. Differences in the labeling mainly result for gaze points in border regions (Section 6), for which it is difficult to decide if they belong to an AOI or not.

3. What is the longest relative time interval with two participants looking at the same AOI?

   AOI_3 from 20%–29%

   This task could be solved with the timeline overview (Figure 13). However, we observed that the interpretation of this view was not clear at the beginning. In combination with this concrete question and a repetition of the explanation from the beginning, the experts claimed that they finally understood how the view works. Hence, the resulting intervals were correct for all experts.

7.2 Questionnaire

We asked the experts to rate the visualization components on a Likert scale from 1 (not helpful) – 6 (very helpful) with the option to give no rating. The questionnaire also contained free-text questions about the used strategies to solve the task and suggestions for improving the visualization and the analysis process.

Table 1 shows the results of the questions about the visualization components. The overall system was rated very useful; especially the cluster editor turned out to be the most useful component to solve the given tasks. All experts stated that they can imagine using our technique for their experiments—except for one expert who stated that their experiments are very standardized and therefore our technique would not be directly applicable. However, some of the components were less used by some of the experts. Two experts did not rate the video player since they did not use it except for some initial testing. One expert rated the timeline overview as not helpful, since the task could also be solved with the other visualization components. Another expert rated the cluster view as less helpful, since the complete time for the annotation was spent in the cluster editor.

From the free-text comments, we could derive additional suggestions to improve the usability and the visual representation. In general, the need for more convenient interaction techniques in the cluster editor was stated by most of the experts. For example, the experts missed hot-keys to interact quickly with the editor and order the clusters individually. In our current implementation, the clusters are always sorted in decreasing order of their total duration. Two experts mentioned that the timeline overview might be replaced by a Gantt chart [13], since people might be more familiar with such a representation. Two experts also mentioned that the main cluster view contained too much information. Since the segments of a cluster were also represented in the editor, they stated that the left part of the visualization (Figure 5 (a$_1$))

Table 1: Answers to the question: How useful was the visualization component? 1 (not helpful) – 6 (very helpful).

Visualization Component	Mean	Standard Deviation
Cluster View	4.5	1.5
Scarf Plot	5.8	0.4
Timeline Overview	4.3	1.7
Cluster Editor	6.0	0.0
Video Player	5.3	1.0
Overall System	5.6	0.5

was not important to them and could be removed from the visualization. One expert missed the information to which video a thumbnail belongs to. As a suggestion, another label on the thumbnail showing the video ID could be included.

7.3 Applied Strategies

In order to solve the task, the experts applied different strategies. Given the set of described possibilities, we could identify the following labeling and analysis strategies:

Video Investigation: The video stimulus was relevant to the experts in two situations: for the initial search for the AOIs and for the interpretation of thumbnails in context of the video. Although it was possible to perform search queries by drawing AOIs in the video (like in the traditional analysis approach), the experts used this function just at the beginning of the task. The main purpose of the video player was to investigate the context of a segment. The experts often selected one of the segments and looked at it in context of the whole video image. Typically, only a couple of consecutive frames were investigated for ambiguous gaze point positions. Except during the initial demonstration, the video player was not used to play longer timespans of one of the involved videos.

Segment Similarity Search: Experts using mainly the cluster editor picked images from the clusters to search for similar images in all other or unlabeled clusters. This was usually performed by searching for specific thumbnails of a labeled cluster, either very similar or very dissimilar to the current representative. Query results were then investigated and corresponding thumbnails that belonged to the searched AOI, as well as unlabeled segments that belonged to one of the other AOIs were labeled in parallel. Searching for similar thumbnails just in unlabeled clusters results in an iterative reduction of the set of thumbnails that have to be investigated.

Sequential Cluster Browsing: One expert followed the systematic approach to select each cluster after the other to either decide if its content belongs to one of the AOIs or can be removed from the data. All irrelevant clusters were placed in the garbage can. This approach was also followed by Annotator 2 in the use case (Section 6). Although it might seem costly to have to look through all clusters and images, each image is typically investigated only once. In tasks were every segment requires a label and not only a subsection of the images needs to be labeled, this approach can be very efficient. This approach requires the analyst to know the AOIs and which segments can be discarded, which is typically the case in hypothesis-driven experiment settings.

Scarf Plot Annotation: One expert mainly focused on identifying long segments in the scarf plot. By selecting one of the unlabeled segments, the editor showed the corresponding cluster. Labeling and correcting this cluster led to the colorization of the respective segments in the scarf plot showing other time spans with attention on this specific AOI. With this approach, the annotation time is reduced by focusing on the most relevant long segments first. In many cases, small segments of the same AOI as the investigated long segment are labeled as well, without the analyst having to look at them again.

In summary, the experts applying the last two strategies were the most efficient ones. In a thorough analysis scenario, we would suggest identifying all relevant AOIs at the beginning and then proceed through the clusters or scarf plot to either assign the correct labels or mark the thumbnails as irrelevant for the analysis.

8 Conclusion

We presented a new visual analytics approach for mobile eye tracking data based on automatic image comparison and clustering. Our technique provides eye tracking experts with an overview of the distribution of attention on different AOIs, even for multiple videos with unconstrained conditions from different participants. Our use case showed that the annotation with our technique is far more efficient than the common state-of-the art approaches based on dynamic AOIs. As a consequence of this efficiency gain, we suggest improving the annotation results by letting multiple annotators label the data. Issues in the inter-annotator agreement can then be checked by looking at ambiguous segments again. This approach could be integrated in the cluster editor by adjusting the borders of the ambiguous thumbnails with the colors of the different AOIs they have been assigned to. The user could then filter for the most ambiguous elements and decide where they belong to.

We identified the main shortcoming of our approach in the interpretation of gaze points in border regions of AOIs. In these cases, single representative images might not be sufficient to judge if a segment belongs to an AOI or not, and the corresponding timespan in the video has to be investigated. One possible solution to this issue could be achieved by representing the complete image sequence of a segment, as described by Kurzhals et al. [20]. Hence, an additional view, or an integration into the scarf plot (e.g., by showing thumbnails at a specific zoom level) depicting the sequence would be possible.

Because of the user-centered design of our approach, the main focus of this work lies on the analysis of hypothesis-driven experiments with predefined AOIs. Another application of mobile eye tracking deals with unconstrained scenarios to capture natural behavior (e.g., making a sandwich [14]) where interesting areas have to be discovered during the analysis. With our approach, clusters could be identified if different participants attended to the same objects. If no attention was spent on potentially interesting objects, our current approach would exclude this data. With more specific knowledge about the environment, our approach could also be adapted to include clusters where no attention was spent on a potentially interesting area. This would require additional visual coding of these elements. In general, our approach could be applied to any time-dependent image series to identify and label similar content, provided that the applied similarity metric is appropriate for the comparison.

For future work, we plan to extend our approach to long-term experiments, such as car-driving scenarios, where long video sequences with some static (i.e., the dashboard elements) and some highly dynamic AOIs have to be analyzed. This is especially challenging since the dynamic content (e.g., a short moment of inattention) can be hard to identify in the recorded data. We also plan to conduct a user study with participants of different experience levels to further evaluate the applicability of our approach. To further improve the scalability of our technique, we plan to extend the technique by an interactive classification component. When a sample of the recorded data has been labeled, the applied bag-of-features approach can directly be used to train a support-vector-machine (SVM) classifier with the labeled segments as positives and the other clusters as negative samples. Time segments from new participants could then be analyzed by the trained classifiers and depicted in the editor before assigning them to the appropriate labels.

Acknowledgments

We thank the German Research Foundation (DFG) for financial support within project B01 of SFB/Transregio 161.

References

[1] W. Bailer and G. Thallinger. A framework for multimedia content abstraction and its application to rushes exploration. In *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*, pages 146–153, 2007.

[2] M. Banerjee, M. Capozzoli, L. McSweeney, and D. Sinha. Beyond kappa: A review of interrater agreement measures. *Canadian Journal of Statistics*, 27(1):3–23, 1999.

[3] P. Bertolino. Sensarea, a general public video editing application. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 3429–3431, 2014.

[4] A. Bhattacharyya. On a measure of divergence between two multinomial populations. *Sankhyā: The Indian Journal of Statistics*, 7:401–406, 1946.

[5] 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-STARs*, pages 63–82, 2014.

[6] R. Borgo, M. Chen, B. Daubney, E. Grundy, G. Heidemann, B. Höferlin, M. Höferlin, H. Leitte, D. Weiskopf, and X. Xie. State of the art report on video-based graphics and video visualization. *Computer Graphics Forum*, 31(8):2450–2477, 2012.

[7] G. Brône, B. Oben, and T. Goedemé. Towards a more effective method for analyzing mobile eye-tracking data: Integrating gaze data with object recognition algorithms. In *Proceedings of the 1st International Workshop on Pervasive Eye Tracking & Mobile Eye-based Interaction*, pages 53–56, 2011.

[8] N. A. Chinchor, J. J. Thomas, P. C. Wong, M. G. Christel, and W. Ribarsky. Multimedia analysis + visual analytics = multimedia analytics. *IEEE Computer Graphics and Applications*, 30(5):52–60, 2010.

[9] O. de Rooij and M. Worring. Active bucket categorization for high recall video retrieval. *IEEE Transactions on Multimedia*, 15(4):898–907, 2013.

[10] A. Duchowski. A breadth-first survey of eye-tracking applications. *Behavior Research Method, Instruments, & Computers*, 34(4):455–470, 2002.

[11] Y. Fu, Y. Guo, Y. Zhu, F. Liu, C. Song, and Z.-H. Zhou. Multi-view video summarization. *IEEE Transactions on Multimedia*, 12(7):717–729, 2010.

[12] M. Furini, F. Geraci, M. Montangero, and M. Pellegrini. Visto: Visual storyboard for web video browsing. In *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*, CIVR '07, pages 635–642, 2007.

[13] H. L. Gantt. *Work, Wages, and Profits*. The Engineering Magazine Co., 1913.

[14] M. Hayhoe and D. Ballard. Eye movements in natural behavior. *Trends in Cognitive Sciences*, 9(4):188–194, 2005.

[15] K. Holmqvist, M. Nyström, R. Andersson, R. Dewhurst, H. Jarodzka, and J. Van de Weijer. *Eye Tracking: A Comprehensive Guide to Methods and Measures*. Oxford University Press, 2011.

[16] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 604–610, 2005.

[17] T. Kobayashi, T. Toyamaya, F. Shafait, M. Iwamura, K. Kise, and A. Dengel. Recognizing words in scenes with a head-mounted eye-tracker. In *Proceedings of the IAPR International Workshop on Document Analysis Systems (DAS)*, pages 333–338, 2012.

[18] K. Kurzhals, F. Heimerl, and D. Weiskopf. ISeeCube: Visual analysis of gaze data for video. In *Proceedings of the ACM Symposium on Eye Tracking Research & Applications (ETRA)*, pages 43–50, 2014.

[19] K. Kurzhals, M. Hlawatsch, M. Burch, and D. Weiskopf. Fixation-image charts. In *Proceedings of the ACM Symposium on Eye Tracking Research & Applications (ETRA)*, pages 11–18, 2016.

[20] K. Kurzhals, M. Hlawatsch, F. Heimerl, M. Burch, and D. Weiskopf. Gaze stripes: Image-based visualization of eye tracking data. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):1005–1014, 2016.

[21] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 1150–1157, 1999.

[22] H. Luo, J. Fan, J. Yang, W. Ribarsky, and S. Satoh. Exploring large-scale video news via interactive visualization. In *IEEE Symposium on Visual Analytics Science And Technology*, pages 75–82, 2006.

[23] T. Munzner. A nested model for visualization design and validation. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):921–928, 2009.

[24] R. Netzel, M. Burch, and D. Weiskopf. Interactive scanpath-oriented annotation of fixations. In *Proceedings of the ACM Symposium on Eye Tracking Research & Applications (ETRA)*, pages 183–187, 2016.

[25] T. Pfeiffer, P. Renner, and N. Pfeiffer-Leßmann. EyeSee3D 2.0: Model-based real-time analysis of mobile eye-tracking in static and dynamic three-dimensional scenes. In *Proceedings of the ACM Symposium on Eye Tracking Research & Applications (ETRA)*, pages 189–196, 2016.

[26] D. F. Pontillo, T. B. Kinsman, and J. B. Pelz. SemantiCode: Using content similarity and database-driven matching to code wearable eyetracker gaze data. In *Proceedings of the ACM Symposium on Eye-Tracking Research & Applications (ETRA)*, pages 267–270, 2010.

[27] 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.

[28] O. Rooij, J. van Wijk, and M. Worring. Mediatable: Interactive categorization of multimedia collections. *IEEE Computer Graphics and Applications*, 30(5):42–51, 2010.

[29] J. Schöning, P. Faion, and G. Heidemann. Pixel-wise ground truth annotation in videos – an semi-automatic approach for pixel-wise and semantic object annotation. In *Proceedings of the International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, pages 690–697, 2016.

[30] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

[31] T. Toyama, T. Kieninger, F. Shafait, and A. Dengel. Gaze guided object recognition using a head-mounted eye tracker. In *Proceedings of the ACM Symposium on Eye Tracking Research & Applications (ETRA)*, pages 91–98, 2012.

[32] H. Y. Tsang, M. Tory, and C. Swindells. eSeeTrack–visualizing sequential fixation patterns. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):953–962, 2010.

[33] T. Yue, Y. Dai, and Y. Liu. A hue-saturation histogram difference method to vehicle detection. In *International Conference on Multimedia Technology (ICMT)*, pages 31–34, 2011.

[34] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems 17*, pages 1601–1608, 2004.