

CVR-Analyzer: A Tool for Analyzing Cinematic Virtual Reality Viewing Patterns

Sylvia Rothe
LMU Munich
Munich, Germany
sylvia.rothe@ifi.lmu.de

Tobias Höllerer
University of California
Santa Barbara, United States
holl@cs.ucsb.edu

Heinrich Hußmann
LMU Munich
Munich, Germany
hussman@ifi.lmu.de

ABSTRACT

Cinematic Virtual Reality has been increasing in popularity over the last years. Watching omnidirectional movies with head mounted displays, viewers can freely choose the direction of view, and thus the visible section of the movie. In order to explore the users' viewing behavior, methods are needed for collecting and analyzing data. We developed an analyzing tool, the CVR-Analyzer, which can be used for inspecting head pose and eye tracking data of viewers experiencing CVR movies. The visualized data are displayed on a flattened projection of the movie as flexibly controlled augmenting annotations, such as tracks or heatmaps, synchronously with the time code of the movie and allow inspecting and comparing the users' viewing behavior in different use cases.

Author Keywords

Cinematic Virtual Reality, 360° movie, omnidirectional movie, eye tracking, head pose, heatmaps.

ACM Classification Keywords

H.5.1 Multimedia Information Systems: Artificial, augmented, and virtual realities; I.2.10 Vision and Scene Understanding: Video analysis

INTRODUCTION

Omnidirectional movies are attracting widespread interest and have many possible applications, e.g. telling stories about exciting locations in the world, or documenting places of historic interest. Even though the term 360° video is widespread, it does not accurately reflect this media. On the horizontal level, there are indeed 360° to explore, however, in full-surround videos, there are also vertical angular extents of $\pm 90^\circ$ to be observed. We use the term omnidirectional video.

In Cinematic Virtual Reality (CVR) the viewer watches omnidirectional 2D or 3D movies using a head-mounted display or other VR devices. Thus, the viewer is positioned inside the scene and can freely choose the direction of view. Accordingly, the viewer determines the visible section of the

movie, and so it may happen that details, important for the story, are missed. For developing new methods of attention guiding, we need knowledge where the viewers look, if they follow any cinematic cues, such as sounds and movements, or stimuli, such as flicker highlights and guiding messages. The necessary research requires methods for recording and analyzing head and eye movements.

Regarding the viewers' watching behavior, it is often desirable to identify areas where hotspots are forming and to compare hotspot areas for various use cases. The term hotspot is used in the literature in different ways. In our work, a hotspot is an area with high numbers of track points in a certain space-time segment.

Users' head-orientation and eye gaze behavior can be displayed as virtual trails or accumulated as heatmaps on top of the surround-view backdrop. Heatmaps, as well as track visualizations, are well-established visualization concepts. For CVR data sets, we need dynamic heatmaps, because data and stimuli are changing over time. When the omnidirectional scene is static, the heatmap overlay can persistently accumulate, using arbitrary trail life-time intervals. When the omnidirectional scene changes, heatmap accumulation should be reset, and when the surround-video is in motion, the visualizations should be fleeting or should be accumulated on top of the scene.

Some commercial products [27–29] provide analytics for movies. The video heatmaps of Wistia [28] offer contextual information (when and where the video was watched). In a color-coded timeline is represented how the viewer interacts with the video, which parts of the video were skipped, which parts were seen and how often. This is coded by color and also called heatmap. In contrast, we are interested in the viewing direction of the user, which we visualize at the movie frames. Viar360 [27] offers heatmaps on movie pictures. However, the visualized data do not show the viewing direction. Producers can define PoI and the heatmap shows how often users activate them. Because omnidirectional videos are also used for advertising, producers are interested in the information, where the user is looking. Hence more and more such platforms are arising, we could not inspect all of them. Some tools need a minimum number of views for generating heatmaps: youtube heatmaps [30] at least 1000 views, facebook [31] more than 50,000 unique viewers. Indeed, we have fewer participants in our studies. In such cases it needs the possibility to adapt

various parameters to the testing situation. All these products assume that the videos are hosted to the providers platform. There are several missing features, which we need in our research. Summarizing, there are the following reasons for developing the CVR-Analyzer:

- Movies and data should not be on foreign servers.
- Commercial tools need a large number of participants.
- Adjustments of specific parameters are needed for smaller numbers of participant.
- Missing features in commercial tools are:
 - tracks for each participant,
 - excluding participants, separating participants,
 - comparing groups,
 - investigating eye tracking data,
 - precomputed hotspot overview.

In addition, an own tool offers the opportunity to react quickly to new requirements.

RELATED WORK

Much research in recent years has focused on attention guiding in Cinematic Virtual Reality. Pausch et al. [20] examined how the attention of the viewer can be drawn to a desired spot. The orientation of the head was logged, and conventional histograms were used for illustrating the data. Similarly, Sheikh et al. [4] connected several cues (motion, gestural and audio cues) to the main character of a scene. The head orientation was recorded and the percentage of people who had seen the target over time was evaluated. Furthermore, the results were displayed via diagrams depicting the times a target was seen. In histograms, tables and diagrams, specific values were presented. With our approach, we want to illustrate the viewing behavior by visualization the data as augmentation on top of the movie. Nielsen et al. [19] compared an attention-grabbing moving firefly with forced rotation and no guidance. Questions were used for finding out which story elements attracted the attention of the viewers. Our tool is focused on analyzing eye and head tracking data (the middle of the viewport), providing insights into which cues catch the attention of the viewer and which can change the viewing direction.

Related research investigated head and eye movement datasets for omnidirectional images [1,16,21,25]. The obtained results about saliency maps are useful for rendering, transmission and compression algorithms [1,21]. Upenik et al. [25] described a method to obtain fixation locations and continuous fixation maps. In our studies, the images are varying continuously over time, so the additional time component needs to be accounted during analysis.

Head and eye tracking data are spatiotemporal data. Netzel [18] introduced Hilbert attention maps for visualizing spatiotemporal eye tracking data. Hilbert curves were used for mapping the two-dimensional space to one dimension.

However, not all points, which are neighbours in the 2D space, are also neighbours in the 1D space. Furthermore, it is difficult to assign hotspots to the areas in the movie. Corbillon et al. [8] examined a dataset of head movements and presented the statistic results in diagrams and tables. Another approach for visualizing spatiotemporal data is a space time cube [2,10], formed by the two space coordinates and the time coordinate. Since our space data are located on a sphere, we would need to consider projection to a Euclidean plane and areas at the left and right border are separated, even if they are close together in original. Additionally, some effort is needed to reconcile the time slices of the cube with the pictures in the movie.

There are several methods for the visualization of eye tracking data which can also be applied for head tracking data. Blascheck et al. [3] classified various methods and analyzed 110 research papers that used visualization techniques of eye tracking. Different types of stimuli were investigated. Videos are dynamic stimuli, which have not only a space component, but also a time component, so heat maps are changing dynamically. Using motion-compensated maps [15], moving objects can be observed. However, the outcome is a static map without a visualization of the temporal gaze information. Burch et al. [6] introduced the AOI rivers technique for visualizing time-varying frequencies of eye gaze fixations, showing the data as river-like structures. However, for our research it is important to visualize the data directly on the movie.

For inspecting our data in overview fashion on traditional displays, we rely on projection of the sphere to a plane. Many options are available, each distorting space in a different way. Map guides [1], [13] can help to find the most fitting projection. Jiang et al. [12] classified map types and investigated spatial interpolation methods for several types of projections. We inspected various projection and weighed the pros and cons. The equirectangular projection of the 2-dimensional sphere to a Euclidean plane was chosen, since omnidirectional videos are typically formatted in this way and there was no advantage of the other projections.

BACKGROUND

Device

There are several options for displaying the collected data, e.g. directly on the HMD or using other displays.



Figure 1: The reduced field of view in the HMD (left) makes it difficult to inspect the recorded data, since just a small part of the data is visible.

The perspective of a researcher interested in viewing phenomena is different from a CVR consumer's perspective, as overview and across-surround comparisons become important. Thus, exploring the data directly via HMD (Figure 1) quickly proved to be cumbersome, since just a small part of the data is visible at any given time.

Projection

Similar to geographic data, our logged head and eye tracking data have two coordinates on the surface of a sphere, with an azimuth between -180° and 180° , and elevation between -90° and 90° . For visualizing spherical data, they can be projected to a two-dimensional Euclidean space (Figure 2). However, this changes the metric and brings some challenges, such as for example distortion.

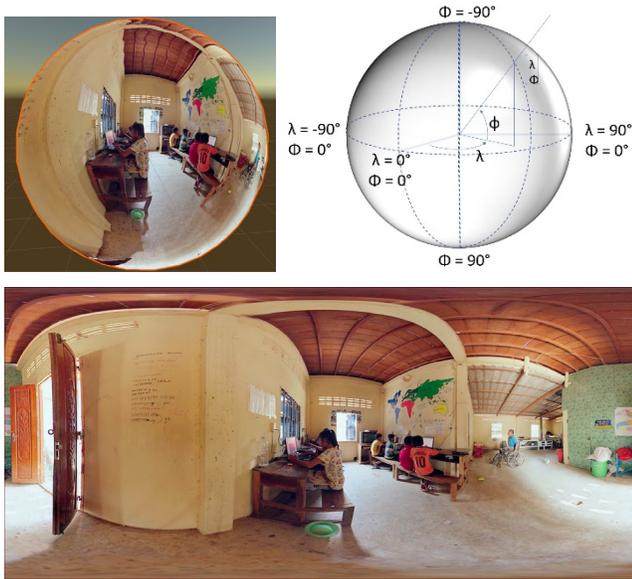


Figure 2: The coordinates of the sphere for an omnidirectional movie (above), equirectangular projection (below).

Heatmaps

The heatmap represents how frequently viewers looked at a certain area segment within a specific time frame (Figure 3).



Figure 3: View of the movie in heatmap mode.

Since viewers will not fixate their attention to the same exact point of the movie, a smoothing filter is needed. We realized that by a small cell around the view point and a radial gradient. For a single isolated view point the heat value at the center of the surrounding cell is a configurable value. The value is called step value, because a further view point with

exactly the same coordinates will increase the heat value by this step value. Depending on the distance of a point inside the cell from the center, the heat value at this point (as well as the opacity) decreases, so that the cell border and the points outside the cell have values of zero. The simplest gradient function is a monotonically decreasing linear gradient:

$$grad(r) = S \begin{cases} \left(1 - \frac{1}{R} r\right) & r < R \\ 0 & r > R \end{cases},$$

where S is the step value, R the cell radius on the sphere. For computing the heatmap on the Euclidean plane we have to take into account the distortion of the projection which make the gradient function a little bit more complicate (see section Projection and Distortion).

Several other gradient functions are possible to influence the color and opacity changes in a cell (Figure 4).

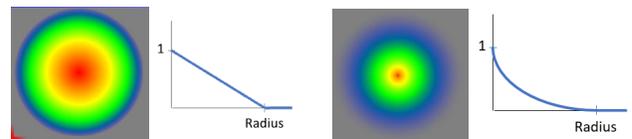


Figure 4: Different gradient functions, linear gradient (left), nonlinear gradient (right)

For points in the intersection of cells the respective heat and opacity values will be added.

$$h_t(x, y) = h_{<t}(x, y) + grad(\sqrt{(x - x_c)^2 + (y - y_c)^2} S),$$

$$h_0(x, y) = 0,$$

where $h_{<t}(x, y)$ is the heat value at the point (x, y) for the seek time less t and $h_t(x, y)$ the heat value at the point (x, y) if a new view point at (x_c, y_c) has to take into account, S is the step value. (Later on, we will consider the case, to restrict track points are used for computing heat points by a life-time interval. Points, which drop out of the specified interval will be subtracted from the current heat value.)

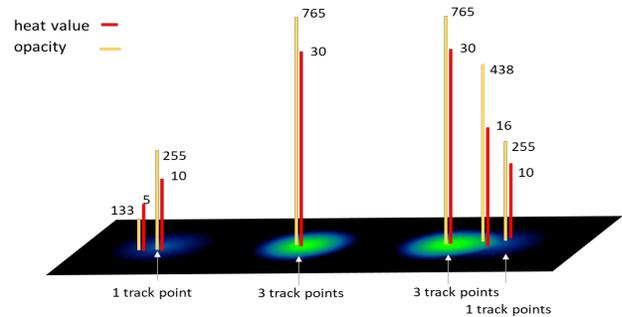


Figure 5: Assuming a step value of 10, the left case shows the heat value and opacity for a cell of a single heat point, the middle case for a triple heat point and the right case for overlapping of cells for a triple and a single heat point.

A similar function is used to compute the opacity at a point. Figure 5 illustrates this graphically.

Finally, the heat color is a mapping of the heat value to a color scale (Figure 6), e.g.

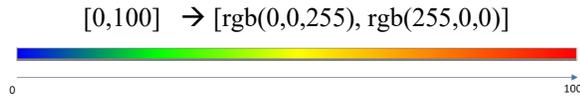


Figure 6: Color scale, a value of 50 corresponds to yellow, a value of 100 to red.

CHALLENGES

In our previous research and in discussions with researchers in Cinematic VR we identified several requirements and challenges for analyzing CVR spatiotemporal data.

Availability, Free Data Source, Performance

Mostly used for research studies the tool should be easy to access without any installation, utilized for a variety of purposes by separating the technology of data recording from data analyzing and provide a level of performance to update visualization of the data in a movie frame rate.

Modes

Researchers need to inspect where the areas are that were most frequently watched by viewers (heatmap mode). They would also like to inspect the trace of only one participant or a group of participants (track mode).

Projection and Distortion

Omnidirectional videos are typically formatted in an equirectangular projection of the 2-dimensional sphere to a Euclidean plane:

$$P: \begin{pmatrix} \lambda \\ \varphi \end{pmatrix} \rightarrow \begin{pmatrix} x \\ y \end{pmatrix}, \quad \lambda, x \in [-\pi, +\pi], \quad \varphi, y \in \left[-\frac{\pi}{2}, +\frac{\pi}{2}\right]$$

The viewing direction is a vector (λ, φ) where λ represents the yaw and φ the pitch. We have to take into account that the seamless sphere is projected to a rectangle and points at the lateral edges are neighbors of the opposite side.

This requires to consider, that points at the lateral edges are neighbors of the opposite side and that the projection changes the metric. To avoid relevant areas to be at the edges it should be possible to rotate the view.

Time Dependency

There is a time component in addition to the space. Even if we can generate a static heatmap for every moment in the movie, it must be kept in consideration how long a viewer is watching an area. In contrast to static images, where neighbor points in a 2-dimensional space are considered, CVR data have three dimensions.

Hotspots Overview in Timeline

It is very time consuming to search through a movie for hotspots. For a first overview the tool should provide an opportunity for detecting hotspot areas across the timeline at a glance.

Comparing Data / Various Sources

Spatiotemporal data in CVR can be logged for various sources: head tracking data, eye tracking data, tracks of stimuli (e.g. flicker highlights). Researchers voiced a need for both, inspecting them together, and separating them. Moreover, in our experiments we saw a need to visualize a comparison between certain groups of data information, like head tracking data vs. eye tracking data, or comparing the tracking data recorded for different guiding methods in a user study.

In summary the following main features are needed for analyzing eye- and head tracking data in CVR:

- The logged data can be displayed on the movie in two modes. Heatmaps show where data points form hotspots (Figure 3) and can be computed overall or for dedicated groups of users. The viewing track of every participant can be displayed (Figure 10). Tracks can be inspected separately. Both heatmaps and tracks are displayed as augmenting annotations, synchronously with the time code of the movie.
- In the play mode, the heatmap is dynamically changing and can be watched as a movie. The life-time value can be used for limiting the time the track points are used for visualization (Figure 17).
- The interval mode is useful to inspect the heatmap and tracks around a specific timecode while the playback is paused. It is possible to adjust the interval, the opacity and the color scale.
- In order to identify hotspots, the tool provides the option to pre-compute the local maximum of heat values for each time code in the defined life-time interval (Figure 18).
- Defining groups allows to inspect the grouped data separately and also to compare it to each other (Figure 21).

THE CVR-ANALYZER

While watching a CVR movie, a recording script logs the head and/or eye movements and the movie time code in a readable very easy format. The recorded file includes the movie name, an identifying tag and the coordinates of each track point (Figure 7).

```
movie: movieName
tag: E_20171102143744
time: 2017-11-02_14:37:44

# seekPos x y
0.002 6.397264 275.2098
0.017 6.357021 274.9597
0.034 8.262152 275.1749
...
152.254 22.63615 67.400
152.265 22.76232 67.438
```

Figure 7: Structure of a log file. The file includes some meta information on the head. Any record entry comprises three parameters: the movie time code, the pitch φ , which varies between -90 and +90 and the yaw λ , varies between 0 and +360

For our research we implemented a C# script and used Unity to generate the log files. While watching a CVR movie, the recording script logs the head (the middle of the viewport) and/or eye movements and the movie time code 60 times per second by default. The logging frequency can be adjusted. However, the CVR-Analyzer works unrelated to any specific recording implementation, while the log file has the format as shown in Figure 7.

Availability

For easy access without any need for installation, the tool is implemented in HTML5 using the HTML5 video and canvas / SVG elements. The CVR-Analyzer can be used locally or remotely over HTTP and requires a Web Server (Figure 8). For performance reasons the omnidirectional movie and the recorded data can be provided on the client.

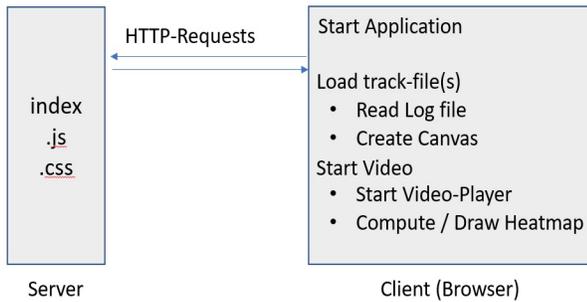


Figure 8: Basic software architecture and data flow.

Starting the CVR-Analyzer, the data of selected log files will be read into memory. While the projected movie is screened, an animation frame is called recursively (on our hardware about 60 times per second). Each call processes the data falling in the time period since the last call and renders the canvas. Depending on the chosen mode, heatmaps or tracks are computed and displayed.

Heatmap and Track Mode

Simple adding a static step value S raises an important issue:

Reaching a heat value corresponding to the maximum value of the color scale, there will no longer be any color changes. To avoid this clamping situation and display the differences beyond this value, the tool provides several options:

- Using a nonlinear gradient function like shown in Figure 4 right reduces this issue, because the maximum color value will be reached only near the center and less in the overlapping areas.
- Scaling the assignment of the heat value to the color has the effect that the maximum color value will only be reached for higher heat values. In this way the differences in the higher value range are recognizable.
- Another possibility is to replace the static step value S by a dynamically one (S'), which depends on the current heat value (Figure 9).

$$S' = \left(1 - \frac{h_{<(x,y)}}{100}\right) S$$



Figure 9: Different step options: a static step value (left) – no color changes behind 100; dynamic step value (right) - the maximum color value cannot be reached.

A second issue. The heat value and therefore the color depends on the number of viewers. Therefore, comparing tests with different numbers of viewers can become challenging. The application normalizes value changes across different numbers of viewers.

Investigating the heatmap, various other parameters can be adjusted, e.g. cell radius, step value, scaling.

In addition to the heatmap mode, the tool offers the track mode and allows researchers to inspect the viewing track of each participant separately (Figure 10).



Figure 10: View of the movie in track mode. A track is shown for every participant and can be disabled and enabled.

Inspecting the data, it can become necessary to exclude the track of a certain participant or a group of participants. We provide the functionality to switch on or off tracks of viewers for the computation of heatmaps as well as for displaying tracks.

Projection and Distortion

Using equirectangular projection for the movie and the logged data has the advantage of seeing the whole picture at once. However, the projection changes the metric: the further an area is away from the horizontal 0° -line, the more the area is distorted (Figure 11).

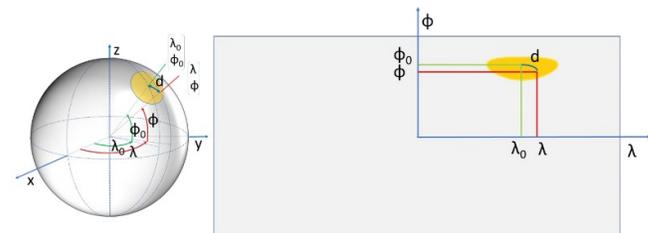


Figure 11: The equirectangular projection of a sphere to a flat plane and the distortion of a cell.

The distance d between two points (λ, φ) and (λ_0, φ_0) is calculated as:

$$d = \left\| \begin{pmatrix} \lambda \\ \varphi \end{pmatrix} - \begin{pmatrix} \lambda_0 \\ \varphi_0 \end{pmatrix} \right\|_{sphere}$$

$$= \cos^{-1}(\sin \varphi_0 \sin \varphi + \cos \varphi_0 \cos \varphi \cos(\lambda_0 - \lambda))$$

Even if the distance between two vertical points (same λ) on the sphere is the same as the distance between its projected points,

$$\left\| \begin{pmatrix} \lambda \\ \varphi \end{pmatrix} - \begin{pmatrix} \lambda \\ \varphi_0 \end{pmatrix} \right\|_{sphere} = |y - y_0|$$

the distance between two horizontal points (same φ) varies:

$$\left\| \begin{pmatrix} \lambda \\ \varphi \end{pmatrix} - \begin{pmatrix} \lambda \\ \varphi \end{pmatrix} \right\|_{sphere} = \cos^{-1}(\sin^2 \varphi + \cos^2 \varphi \cos(\lambda_0 - \lambda))$$

In the manually generated heatmap of Figure 12, all displayed areas have the same size on the sphere. Inspecting such heatmaps, an area can be overestimated on account of their size. So, the user can switch off the distortion. In this case the heatmaps are computed with Euclidean coordinates, which may cause underestimation in the pole regions, since the distance between such points is larger than on the sphere. However, in our own studies, we often benefited from switching between both modes. Besides, most head and eye tracking data in CVR are located fairly close to the 0°-equator-line.



Figure 12: A manually generated heatmap for studying the distortion in the heatmap tool.

While the equirectangular projection is very familiar for CVR movies, and there is already a player for it in HTML5, we also inspected some other projections using Tissot's indicatrix [7][17] (Figure 13 - Figure 15). An indicatrix describes the distortion at a certain point. For some other projections the distortion at the poles is considerably less pronounced, e.g. the Hammer projection. However, there is a distortion near the important 0°-equator-line. In our research the most relevant data are in this area. The equirectangular projection has the advantage to display areas near the 0°-line with only less distortion. Other projection, as the azimuthal equidistant projection (hemispheres), does not seem appropriate, because of the split display. Taking into account that playback for these projections needs more computational effort (it requires to convert each movie frame in real-time) and that most of eye and head pose data occurs

away from the poles, we chose to use the equirectangular projection.

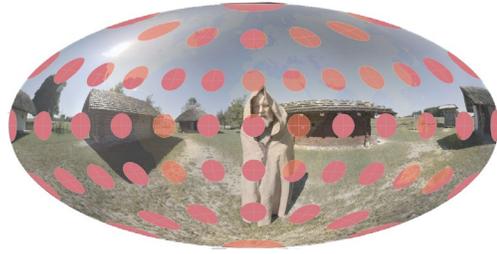


Figure 13: Hammer projection, with overlays according to Tissot's indicatrix. It has the advantage of less distortion in the pole regions, but the disadvantage of more distortion on the sides.



Figure 14: Azimuthal equidistant projection with overlays according to Tissot's indicatrix. It has the advantage of less distortion, but the disadvantage of a split display.

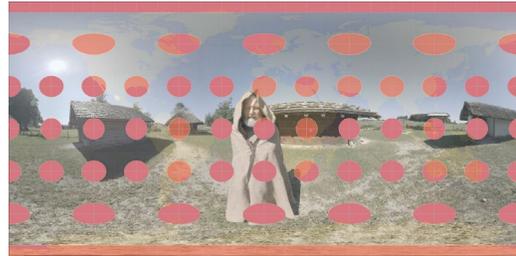


Figure 15: Equirectangular projecting, with overlays according to Tissot's indicatrix. It has the advantage of less distortion in the area around the 0°-horizontal-line, but the disadvantage of more distortion near the poles.

The heatmaps have to be computed assuming continuous surround topology. We continued the cells across the border by transforming the coordinates. The placement of the borders depends on the specific coordinate system in each project, which defines the cut ($\text{yaw} = -\pi$, $\text{yaw} = \pi$). For providing more convenience for inspecting different areas, we added the functionality to rotate the view (Figure 16).

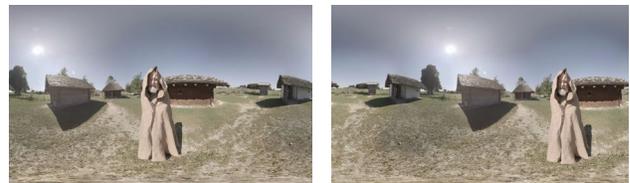


Figure 16: Rotating the view, so relevant areas are not at the edges.

Time Dependency

CVR content is intrinsically dynamic. Tracks and heatmaps are changing over the time. Using the timeline, the data can be inspected for every time code in the movie, either in play mode or interval mode.

In the play mode, the heatmap is dynamically changing and can be watched as a movie. The life-time value can be used to specify the time, the track points are used for computing heat points and their visualization. This parameter is of particular significance in CVR, because meaningful hotspots are formed within only relatively short intervals around a time code (Figure 17).

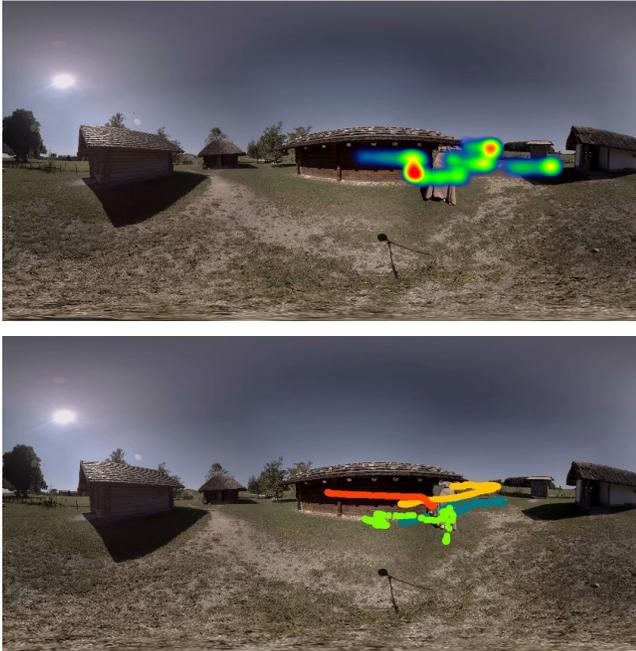


Figure 17: Heatmap (above) and Track (below) presentation using a life-time interval.

The interval mode is useful to inspect the heatmap and tracks around a specific timecode while the playback is paused.

Hotspots Overview

To facilitate the identification of hotspots, we provide the option to pre-compute the local maximum of heat values for each time code, based on a given life-time value.

The local maximum for a time code is the maximum of computed heat values in the life-time interval identified by the time code (Figure 18). By clicking on a particular value in the line chart, the application navigates to the corresponding time code in the movie, displays the image with the heatmap and marks the region of the hotspot (Figure 21).



Figure 18: Hotspot Detection via timeline. The x-axis represents the timecode, the y-value shows the local maximum. In this way, time codes with relevant hotspots can be detected.

Comparing Various Sources

In some cases, the data are grouped naturally, different changing in space and time, but they are all of the same spatiotemporal structure. In our experiments, we had various application scenarios:

- comparing eye and head tracking data
- comparing several guiding methods using the same movie
- comparing tracking data with moving stimuli (e.g. flicker)

A first approach was to compare the different tracks (head pose, eyes movement) of a single participant. This can be done using the track mode and switching off all other tracks (Figure 19).



Figure 19: Eye (green) and head (orange) trace data for one participant, there is a flicker highlight (yellow) on the right side.

However, it reduces the analysis to the data of a single participant, which is not valuable.

Another approach consists in comparing the heatmaps of two groups in separate applications (Figure 20). The disadvantage of using separate windows is the difficulty to compare exact position of the hotspots. On the other hand, it is very clearly to differentiate which hotspot belongs to which group of tracks.

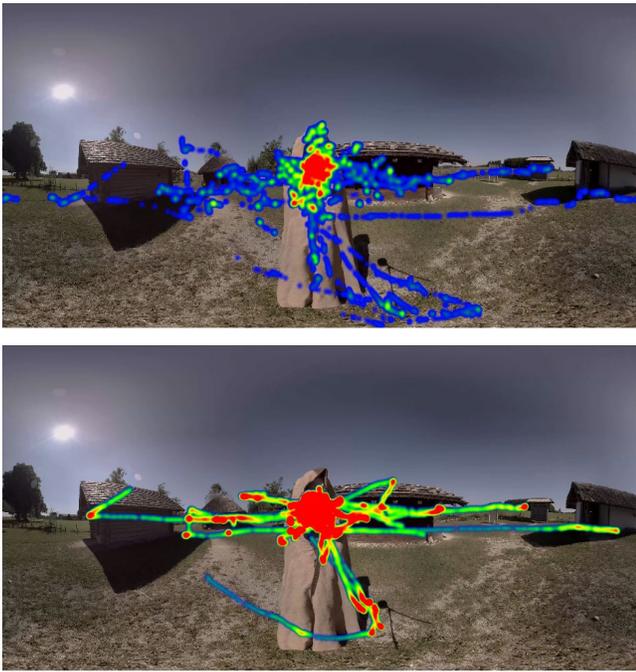


Figure 20: Eye and head tracking data separated by different windows, eye tracking data (above), head pose data (below).

Defining arbitrary groups of track files in one application allows us to apply all the mechanisms described above to groups. In this way it is possible to inspect the grouped data separately and also to compare it to each other.

Currently groups are realized for the hotspot analysis. The line chart gives an overview of the hotspots of the different groups. Furthermore, by clicking on a particular group value in the line chart, the application shows the corresponding to this group heatmap and marks the region of the hotspot with the line color (Figure 22, 23). The coordinates are shown in the status bar. Selecting a chart point with the same timecode of a second group adds the heat areas of this hotspot.

Using the same color scale for different groups in one heatmap, it could be hard to recognize, which heat point belongs to which group. One way is to use monochrome color scales. This gives us the opportunity to display heatmaps of different groups in one frame more illustratively (Figure 21).



Figure 21: Hotspot of groups visualization by monochrome color scales.

As a second way, the tool provides the option to decrease the heat point opacity of one group.

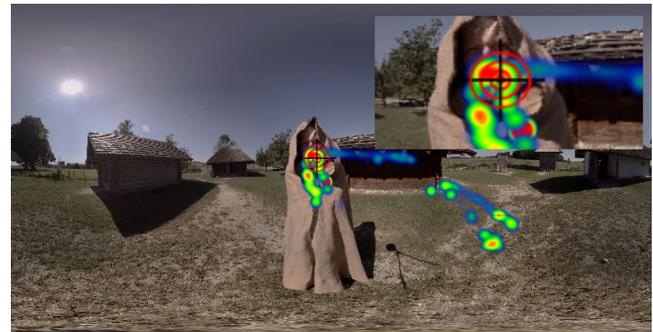


Figure 22: Hotspots of eye (green) and head (red) tracking data are represented by different curves.

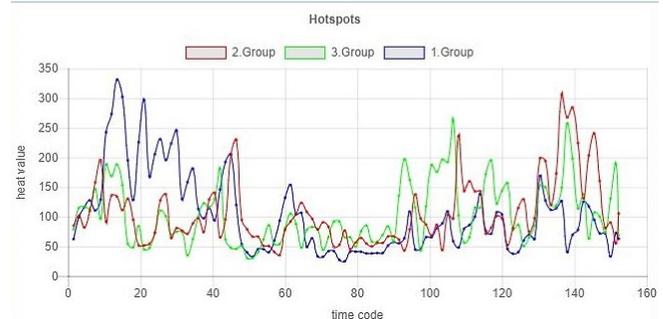


Figure 23: Hotspots of different groups are represented by different curves; the hotspots are marked in the heatmap with corresponding colors.

Performance Considerations

In our experiments, we recorded data from 14 participants in three use cases: head pose, eye movement, and stimuli, like introduced flicker highlights. The logging frequency was 60 times per second. So, we had 2520 data records per second to compute the heatmap. Showing the heatmap as augmenting annotations, synchronously with the time code of the movie raises some performance issues. To provide an attractive visualization even on computers with lower power the tool offers some parameters to speed up the performance:

- Select a lower frame rate for heatmap updates. Per default a canvas will be rendered each time the animation frame is called (on our hardware about 60 times per second). Reducing the canvas update rate reduces the number of canvas updates per second.
- Because the heatmap calculation is pixel based, the computing effort naturally depends on the number of pixels. Normally we use a 1000x500 pixel canvas. This results in a finely structured visualization but in a high computing effort. Using a coarser pixel structure, e.g. a canvas of 500x250 pixel reduces the effort to a quarter.

USE CASES

The CVR-Analyzer has already proved of value in a series of user studies on various research topics in CVR.

User Guidance

A lot of research in the field of CVR have addressed the question how the user can be guided in CVR [4,19,20]. For analyzing the results, it is important to know, if the viewer really followed the cues. This can be found out by questionnaires or by recording head and eye movements. Using our tool for analyzing heatmaps and viewing tracks shows the researcher exactly where the user was looking and supports finding methods for guiding attention [22].

Subtitling in CVR

Since in CVR the viewer chooses freely the field of view, new methods of subtitling are needed [5,32]. It is important to know how the viewer changes the gaze between the movie content and subtitles [23]. Studying the viewing behavior allows to improve methods for subtitles in CVR.

Information Recall

CVR is not only a field for entertainment. For investigating its potential as an aid for education and learning researcher have the know the user's field of view. For traditional film it is obvious what the participant has seen. This is not so in CVR. So, researches need more information to find out why a method works. It is important if an object was never in the field of view or if the user did not recognize it or forgot it. The CVR-Analyzer can assist in this task [24].

Omnidirectional Images

However, the CVR-Analyzer is not only useful for research in CVR. Also, investigating guiding methods for still images it can be important, in which order the user is looking to certain regions. In such cases traditional heatmaps are not

sufficient, it needs dynamical heatmaps changing by timecodes.

Augmented Reality

Similar research can be conducted on Augmented Reality (AR) applications, where the viewing behavior of mobile AR users within their surrounding environment is of high interest to an augmentation designer, and, if surround-view video is captured, or assembled from the user's front-camera views [14], can be analyzed on top of such video material after the fact. While AR use has provided motivating guiding principles for our tool, all evaluations in this paper have been done on conventional CVR content.

CONCLUSION & FUTURE WORK

In this paper, we presented a tool for analyzing spatiotemporal CVR attention data – eye tracking and head pose data logged on a sphere – dynamically changing over time. Important features, which were implemented, are:

- movies and data on the local client
- specific parameters for evaluations of small user groups
- tracks for each participant
- separation of participants
- comparison of groups
- investigation of eye tracking data
- precomputed hotspot overview.

Using the CVR-Analyzer, it is possible to understand more about the users' viewing behavior in CVR and to develop methods for attention guiding.

Our method is usable for various spatiotemporal datasets, especially head and eye tracking data collected in CVR research, but also for static panoramas. Furthermore, the CVR-Analyzer can also be useful for research in Augmented Reality.

Traditional heatmaps show all collected data, independent of the time when a user watched a detail. Where knowledge about the order of watched areas is important, it is necessary to integrate the time component in the visualization. For data in the described format, the CVR-Analyzer provides convenient analysis functionality. To avoid distortion caused by projection to a plane, an inspection of the data in virtual environments using spherical coordinates and head-relative viewing could be a beneficial approach. However, in HMDs the field of view is relatively small, which led to non-optimal analysis options in our experiments. Inspecting the data via walk-in surround-display systems such as a CAVE [9] or the Allosphere [26] could offer a comfortable yet immersive way for investigating such types of data.

The CVR-Analyzer has already proved of value in various research scenarios and assisted in finding results about the viewing behavior in CVR.

REFERENCES

1. Ana De Abreu, Cagri Ozcinar, and Aljosa Smolic. 2017. Look around you: Saliency maps for omnidirectional images in VR applications. In *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*.
2. Benjamin Bach, Pierre Dragicevic, Daniel Archambault, Christophe Hurter, and Sheelagh Carpendale. 2014. A Review of Temporal Data Visualizations Based on Space-Time Cube Operations.
3. T. Blaschek, K. Kurzhals, M. Raschke, M. Burch, D. Weiskopf, and T. Ertl. 2017. Visualization of Eye Tracking Data: A Taxonomy and Survey. *Computer Graphics Forum* 36, 8: 260–284.
4. A. Brown, A. Sheikh, M. Evans, and Z. Watson. 2016. Directing attention in 360-degree video. In *IBC 2016 Conference*.
5. Andy Brown, Jayson Turner, Jake Patterson, Anastasia Schmitz, Mike Armstrong, and Maxine Glancy. 2017. Subtitles in 360-degree Video. In *Adjunct Publication of the 2017 ACM International Conference on Interactive Experiences for TV and Online Video - TVX '17 Adjunct*, 3–8.
6. Michael Burch, Andreas Kull, and Daniel Weiskopf. 2013. AOI Rivers for Visualizing Dynamic Eye Gaze Frequencies. *Computer Graphics Forum* 32: 281–290.
7. Keith C. Clarke. 2017. Map Projections and Coordinate Systems. In *International Encyclopedia of Geography: People, the Earth, Environment and Technology*. John Wiley & Sons, Ltd, Oxford, UK, 1–10.
8. Xavier Corbillon, Francesca De Simone, and Gwendal Simon. 2017. 360-Degree Video Head Movement Dataset. In *Proceedings of the 8th ACM on Multimedia Systems Conference - MMSys'17*, 199–204.
9. Carolina Cruz-Neira, Daniel J Sandin, Thomas A DeFanti, Robert V Kenyon, and John C Hart. 1992. The CAVE: audio visual experience automatic virtual environment. *Communications of the ACM* 35, 6: 64–73.
10. Gautier Drusch, J. M. Christian Bastien, and Stéfane Paris. 2014. Analysing eye-tracking data: From scanpaths and heatmaps to the dynamic visualisation of areas of interest.
11. Bernhard Jenny, Bojan Šavrič, Nicholas D. Arnold, Brooke E. Marston, and Charles A. Preppernau. 2017. A Guide to Selecting Map Projections for World and Hemisphere Maps. . Springer, Cham, 213–228.
12. W. Jiang and Jin Li. 2014. *The effects of spatial reference systems on the predictive accuracy of spatial interpolation methods*. Geoscience Australia, Canberra.
13. John P. Snyder. 1987. *Map projections: A working manual*. Geological Survey Professional Paper, 1395. Washington, DC: U. S. Government Printing Office.
14. Sehwan Kim, Christopher Coffin, and Tobias Höllerer. 2011. Robust Relocalization and Its Evaluation for Online Environment Map Construction. *Visualization and Computer Graphics, IEEE Transactions on* 17, 7: 875–887.
15. Kuno Kurzhals and Daniel Weiskopf. 2013. Space-Time Visual Analytics of Eye-Tracking Data for Dynamic Stimuli. *IEEE Transactions on Visualization and Computer Graphics* 19, 12: 2129–2138.
16. Feng Li, Huihui Bai, and Yao Zhao. 2017. Visual attention guided eye movements for 360 degree images. In *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*.
17. Karen A. Mulcahy and Keith C. Clarke. 2001. Symbolization of Map Projection Distortion: A Review. *Cartography and Geographic Information Science* 28, 3: 167–182.
18. Rudolf Netzel and Daniel Weiskopf. 2016. Hilbert attention maps for visualizing spatiotemporal gaze data. In *2016 IEEE Second Workshop on Eye Tracking and Visualization (ETVIS)*, 21–25.
19. Lasse T. Nielsen, Matias B. Møller, Sune D. Hartmeyer, Troels C. M. Ljung, Niels C. Nilsson, Rolf Nordahl, and Stefania Serafin. 2016. Missing the point. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology - VRST '16*, 229–232.
20. Randy Pausch, Jon Snoddy, Robert Taylor, Scott Watson, and Eric Haseltine. 1996. Disney's Aladdin. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques - SIGGRAPH '96*, 193–203.
21. Yashas Rai, Jesús Gutiérrez, and Patrick Le Callet. 2017. A Dataset of Head and Eye Movements for 360 Degree Images. In *Proceedings of the 8th ACM on Multimedia Systems Conference - MMSys'17*, 205–210.
22. Sylvia Rothe and Heinrich Hußmann. 2018. Guiding the Viewer in Cinematic Virtual Reality by Diegetic Cues. 101–117.
23. Sylvia Rothe, Kim Tran, and Heinrich Hußmann. 2018. Dynamic Subtitles in Cinematic Virtual Reality. In *Proceedings of the 2018 ACM International Conference on Interactive Experiences for TV and Online Video - TVX '18*, 209–214.
24. S. Rothe, F. Althammer, and M. Khamis. 2018. GazeRecall: Using Gaze Direction to Increase Recall of Details in Cinematic Virtual Reality. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia (MUM'18)*.
25. Evgeniy Upenik and Touradj Ebrahimi. 2017. A simple method to obtain visual attention data in head mounted virtual reality. In *2017 IEEE International*

- Conference on Multimedia & Expo Workshops (ICMEW)*, 73–78.
26. Graham Wakefield, Tobias Hollerer, JoAnn Kuchera-Morin, Charles Roberts, and Matthew Wright. 2013. Spatial Interaction in a Multiuser Immersive Instrument. *IEEE computer graphics and applications* 33, 6: 14–20.
 27. Virtual Reality Analytics: Why do you need it and how to use it? Retrieved August 9, 2018 from <https://www.viar360.com/blog/virtual-reality-analytics/>
 28. Heatmaps - Support - Wistia. Retrieved August 9, 2018 from <https://wistia.com/support/stats/heatmaps>
 29. GitHub - Unity-Technologies/360-Video-Heatmaps: Heatmaps for 360 Video. Retrieved August 9, 2018 from <https://github.com/Unity-Technologies/360-Video-Heatmaps>
 30. YouTube Creator Blog: Hot and Cold: Heatmaps in VR. Retrieved August 9, 2018 from <https://youtube-creators.googleblog.com/2017/06/hot-and-cold-heatmaps-in-vr.html>
 31. New Publisher Tools for 360 Video. Retrieved October 8, 2018 from <https://www.facebook.com/facebookmedia/blog/new-publisher-tools-for-360-video>
 32. Exploring Subtitle Behaviour for 360° Video - BBC R&D. Retrieved July 17, 2018 from <https://www.bbc.co.uk/rd/publications/whitepaper330>