Data Maps for Video Interaction Analysis

Nikolas Martelaro
Center for Design Research
Stanford University
nikmart@stanford.edu

ABSTRACT

Video interaction analysis is often used to explore the qualitative interactions and behaviors of people using interactive objects. However, the process of creating quantitative data from video is often time-consuming and tedious work. However, as interactive objects allow data real-time collection, it is possible to use this data in synchronization with video interaction analysis. In this paper, I present a system to use data as a map for exploring qualitative video. Using a sample interaction dataset of a driver experiencing advanced driving features, I show the design development and usage of a tool that links time-series automotive and interaction data with video. The tool provides methods for automatically highlighting and focusing on potentially interesting aspects of the video. Overall, the goal of this tool is to shift the video interaction analysis process from being linearly driven by the video to being driven by the associated data streams in order to reduce analysis time and offer new methods for exploring video.

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous;

Author Keywords

Visualization, video interaction analysis, interaction design

INTRODUCTION

Video is a primary data source in research around understanding people using interactive physical objects. To explore these interactions, video coding tools, such as VCode [6], are used to mark sequence and point events along a video timeline. While this allows one to generate quantitative interaction data from video, it is a tedious process that often involves watching a video many times over to code each specific interaction of interest (ex. every time a person presses a button).

As computation, connectivity, and data collection become more imbued into physical products, new interactive objects can allow for automatic, time-mapped interaction data collection. Although it is easy to create static graphics of these data, it is often difficult to time sync and show these data along with the video. Seeing both data and video at the same time can help alert an analyst to interesting or relevant areas in a video. Additionally, there can often be stretches of video where not much is happening. Using the interaction data logs, it may be possible to skim more quickly over these parts, helping to reduce the time watching uninteresting segments of video. The goal of this work is to create a tool that shifts the focus of video interaction analysis from being video centered to being data centered. The current prototype of this tool is shown in Figure 2. Instead of watching the video linearly and seeing what data occurs or having to mark specific points, the data is used as a control surface for viewing parts of the video.

To move the focus of the video interaction analysis from the video to the data, I explore two areas of visualization, timeline distortion and interactivity. Timeline distortion is explored as a means to provide focus and context to the data charts, which often have large areas of little data during long timelines. By distorting the timeline, it may be possible to reduce the whitespace, giving more screen area to the data and helping the analyst move through the data faster. However, there are challenges that arise from timeline distortion that make the data harder to understand and use as a tool to search the video. The other area I explore is the interactive capabilities of the data. To allow for fast seeking through the video using the data, each data point is a control element. All data points are linked in time to the video. The design of the tool uses interactive focusing and highlighting to reveal to the analyst potential areas of interest to observe in the video.

In the rest of this paper, I describe the dataset used and related work in the area of video interaction analysis. I also provide motivating previous work in visualization that guides the design development of the tool. The implementation is described as well as insights gleaned from the development process and brief testing with people interested in video interaction analysis.

DATASET DESCRIPTION

A user experience study of the driver experience using advanced driving systems is used to create the tool. The study consisted of a person driving a car with automatic lane keeping and cruise control. Video of the drive was recorded as well as associated car data (speed, intelligent cruise control (ICC) on/off, ICC enabled, ICC set speed, brakes). In addition to the car data collection, I employed a Wizard-of-Oz protocol [2] and had a remote automotive designer asking the driver

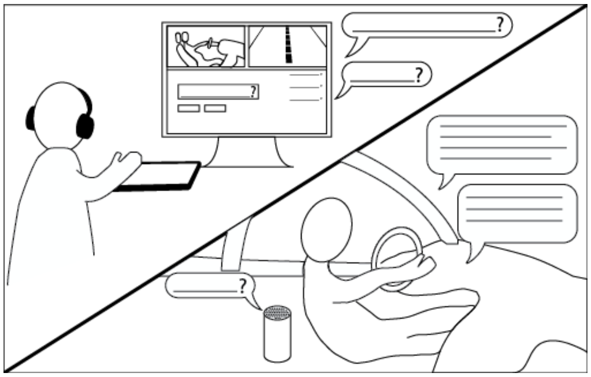

Figure 1. Setup of the remote user experience study. A remote wizard speak to a driver on the road while experiencing new driving systems. Video, car data, and message from the wizard are used as a sample dataset to build the video interaction analysis too.

questions through a text-to-speech system. This remote wizard was provided a live video and data feed from the car and could ask the driver questions in real-time. Figure 1 shows how the remote designer was able to see the driver's interactions on a control station and could send messages to the driver. Often, these designers asked qualitative questions about the driver's experience using the advanced driving features. All data and message from the wizard were logged and time-stamped on a remote server.

RELATED WORK

Video interaction analysis is a common activity for studying and understanding human behavior [8]. This analysis technique allows for quantitative and qualitative data to be gleaned from video through coding specific events. Two common tools to support this are VCode [6] and ChronoViz [3]. While VCode only allows for a viewer to manually tag information on a timeline, ChronoViz can import time-coded data to show in timelines under the video. This allows analysts to leverage data collected during a study to support viewing of the video. In many ways, the data streams become a map for viewing the video and can suggest important areas of the videos to explore. However, the viewer must still manually seek out interesting events in the video and must usually work on a single zoomed timeline that scrolls across the screen. Often, this is a tedious and time consuming task and may not provide the viewer with the best view of the data streams.

For this project, I am interested in using Focus+Context displays [5] to aid in the process of exploring video interactions. I take inspiration from Argawala and Stolte's generalized route maps [1] as they allow for detailed views of maps in areas of high driving interaction (lots of turns) and compress areas of low interaction (long stretches of freeway driving). These visualizations provide salient information about the areas where people are most interested in the information on the visualization. Additionally, using fish eye style distortions [4] to focus in on specific areas of the data, we can automatically distort the timeline view, allowing for more focused views of the data, especially when data become very dense. While fisheye style distortion is one method to view the data, other methods such as CloudLines [10] may also be appropriate for certain data types.

These focus and context techniques can be applicable to long time-series data and may help in highlighting areas areas of high interaction (based on the data) while also providing a general understanding of the overall data picture. Within this area, SignalLens [9] is one of the best examples of focus and context views for long time series data, specifically for electronic signals. SignalLens helped to give the viewer both context and focus of the data, and could highlight areas such as anomalies. One way I extend upon this this work is to have the data linked to video, such as in the ChronoViz system. This allows the user to select data and seek to that point in the video.

Overall, the aim of this project is to have a system which can automatically identify points of interest in a video based on interaction data and focus in on those views while still providing the viewer the overall context for their own exploration of the data. Bringing together automatic focusing and highlighting to data linked to the video may allow for better navigation and exploration of the video data.

METHOD

Implementation

The visualization tool is implemented using HTML, CSS, and JavaScript. D3 [1] is used to create the data charts and handle interaction with the data. JQueryUI [2] is used to create and interact with standard HTML elements such as sliders. The video is hosted on YouTube and embedded into the page using an `iframe`. The YouTube API [3] is used to scrub through the video and access data such as the current video time.

Visual Design

Building upon the visual style of tools such as VCode [6] and ChronoViz [3], the visualization is designed to show the video prominently at the top of the screen, provide individual, plotted data streams below the video, and filter or highlighting controls to the side of the data streams. This provides data analysts a familiar layout and interaction model and can help build upon the analyst's previous work behavior.

Data Charts

Separate colors are used to encode different data streams, aiding the analyst in quickly identifying what data each plot represents. Line charts are used to plot continuous variables, such a vehicle speed. The aspect ratio of line charts are set such that they show more macro trends across time. The aspect ratio for the speed chart is $4.8 : 1$. This aspect ratio also allows for multiple streams of data to fit on a single page view of the entire interface. A 1-D histogram of circles is used for event-based data such as the text-to-speech questions asked by the remote wizard. Each circle is set with an opacity of 0.1 so that overlapping data points become more saturated, highlighting areas of clustered speech events from the remote

[1] https://d3js.org
[2] https://jqueryui.com
[3] https://developers.google.com/youtube/iframe_api_reference

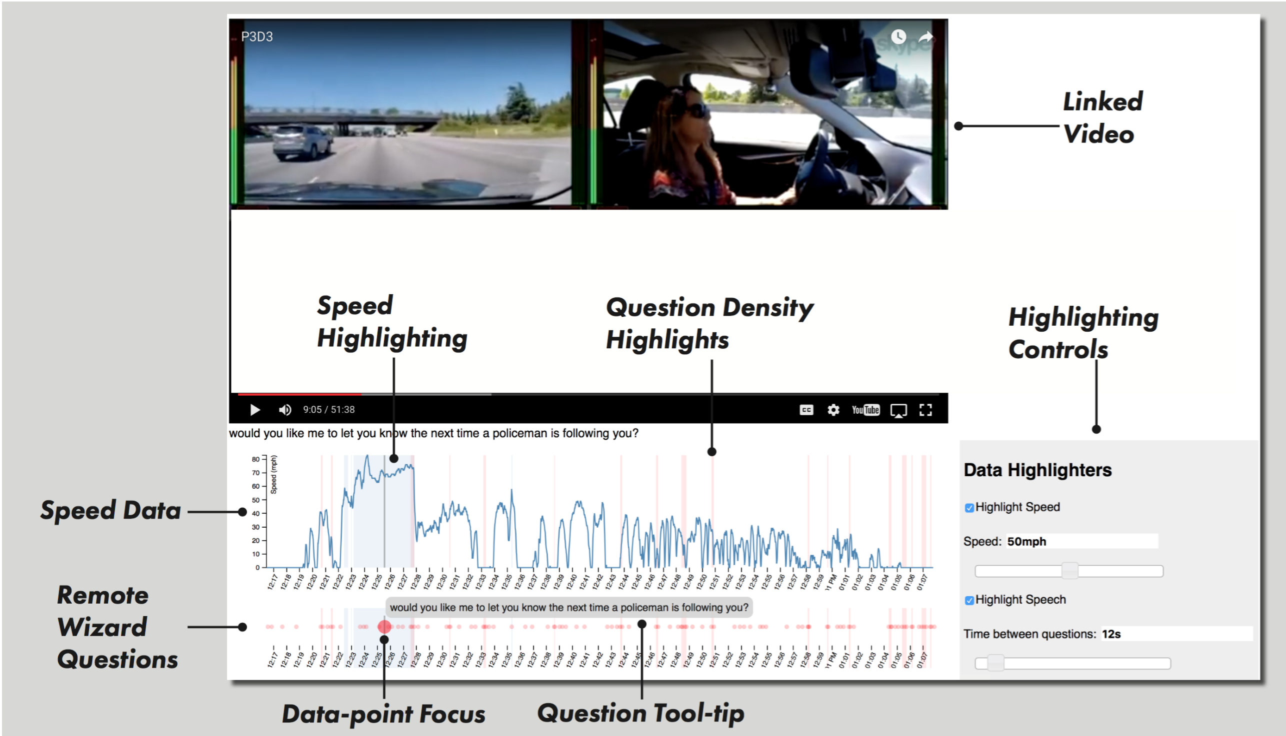

Figure 2. Current interface for the video interaction analysis tool.

wizard. A vertical, gray line is drawn on each chart as a playhead and shows the current position of the video. This playhead is dynamically updated as the video plays and as the analyst moves through the video.

All data are plotted on linear time scales. While this is the obvious scaling, it does give the most accurate representation of the data and is simple for the analyst to interpret. A linear time scale also clearly indicates the frequency of events as they happen in the video. Ticks marks are placed every minute to provide detailed understanding of when in the drive an event occurred.

Although accurate, a linear time scale for event based data can often show large areas of white space, indicating areas of low interaction within the video while causing areas of high interaction to become occluded. Taking inspiration from Agrawala and Stolte's geographically compressed route maps [1], I attempted to reduce the large areas of white space within the 1-D data plots so that it might be easier for an analyst to see moments of high interaction more easily while also retaining a holistic overview of the data.

Alternative Time Axis Formatting

I implemented and explored four different time-axis scales during the development of the visualization. Including the standard linear time scale, I implemented three distorted time scales as shown in Figure 3. These distortions were developed to explore data views that could give both focus and context of the data without becoming too difficult for the analyst to understand. Additionally, each point is also a control interface that can be clicked in order to seek to that time in the video. During the design development of the interface, it became very important for each point to be easily selected. Each time axis scaling is discussed below:

Linear time scale

The linear time scaling (Figure 3-1) is the standard scaling for viewing time-series data. The time scale gives an accurate view of the data and shows moments of high interaction based on the density of data points. The linear time scale is created using a D3 time scale with a domain set to the first and last points of the data, accessed using the `extent` command, and a range defined by the width of the chart.

```
//Width of chart
x.d3.time.scale()
  .range([0, width]);
//First + last data points
x.domain(d3.extent(data, function(d) {
  return d.time;
}));
```

Although the 1-D histogram with translucent circles allows for automatic highlighting of moments with many questions, the linear time scale also occludes points that are overlapping. This can make it difficult to find and click specific points in the data. I found that aside from seeing the overall patterns in the data, the ability to click on points was the most important aspect of the design. With this in mind I attempted to make each point easy to see and click.

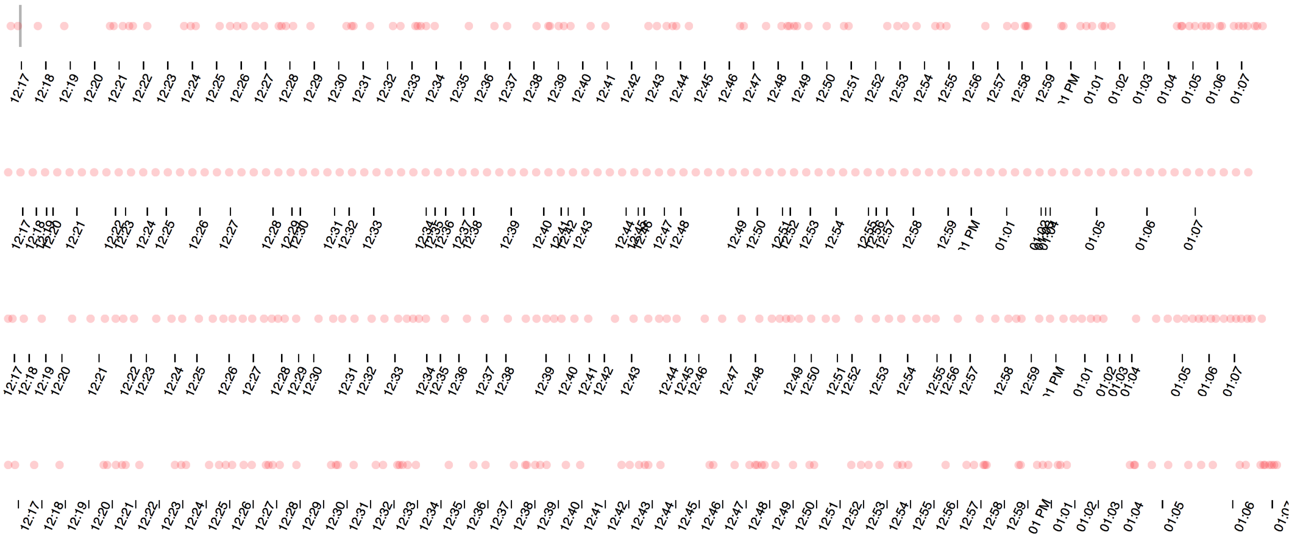

Figure 3. Four alternative time-axis scalings. 1) Linear, 2) Equal point spacing, 3) Smooth backspaced spacing, and 4) Dynamic x-axis fish-eye

Equal spacing of data points scale

This chart (Figure 3-2) distorts the time scale so that each data point was equally spaced using a polylinear time scale [4]. To create the distorted time scale, the domain of the scale is an array of time data for all the data points. The range is then set by creating an array of equally spaced numbers corresponding to the entire width of the chart.

```
numpts = data.length;
step = width/numpts;

//equally spaced range
equalscale = d3.range(0,width,step);

//all time points
timedata = [];
data.forEach(function (d) {
  timedata.push(d.time);
});

// Polylinear time scale
var polylinearTimeScale = d3.time.scale()
  .domain(timedata)
  .range(equalscale);
```

This equally spaced data allows for easy selection of individual points as there is no occlusion. Instead of the data points showing the patterns and density of questions asked, the time scale axis shows this information. The tick marks on the time scale show the inverse of the data density. As the space between each tick mark denotes one minute of time in the video, spread out times show a higher density of questions asked, whereas tightly spaced tick marks denote low density of data. Although this design is quite successful as a control interface, with each point easy to select, it is harder for the analyst to understand where in the data information is dense. The inverse mapping of tick spacing to data density is hard to remember and requires more thought on the part of the viewer and can lead to misinterpretation of the data.

[4] http://bl.ocks.org/herrstucki/5403383

Backspaced smoothing time scale

The backspaced smoothing time axis (Figure 3-3) is scaled such that the distances between points is decreased, reducing the amount of white space between points. Points are moved back proportional to the space between adjacent points ($\propto$ 1.8 in this example). This allows for the chart to be more data dense without removing all the patterns from the data. To distort the axis, the spacing between the current point and the previous point in the shifted space scale is computed. The previous point in the shifted scale is used so that if the previous point had already been shifted back, the current point would take the new position into consideration. If the spacing between the points is greater than half the radius of the point it is shifted back. If the point is too close to the previous point it is left in place so as not to overlap the previous point. The array of shifted points is then used as the range and the time data is used to create the new polylinear time scale.

```
shiftscale = [0];

for (var i = 1; i < data.length - 1; i++) {
  backtimeDiff = (data[i].time -
    data[i-1].time)/1000;
  // compute spacing based on the previous
  // shifted point so that
  backtimeSpace = (x(data[i].time) -
    shiftscale[i-1]);

  // move point back if far enough away
  if (backtimeSpace > radius/2) {
     shiftscale.push(x(data[i].time) -
       backtimeSpace/1.8); //Scaling factor
    }
```

```
  // leave in place if too close
  else {
      shiftscale.push(x(data[i].time));
    }
}

data.forEach(function (d) {
  timedata.push(d.time);
});

// Polylinear time scale
var polylinearTimeScale = d3.time.scale()
  .domain(timedata)
  .range(shiftscale);
```

This shifted scaling does make point selection easier while still retaining some of the overall patterns of the data. Depending on the scaling factor used, the chart will be closer to the linear time scale (scaling factor approaches ∞) or the equally spaced time scale (scaling factor approaches ≈ 1.3). Beyond 1.3 the chart becomes over compressed and smaller than the width of the chart. During testing, 1.8 was found to be a nice scaling factor.

Dynamic x-axis fish-eye focusing

This scale (Figure 3-4) uses a simple, dynamic, Cartesian fish-eye focus to spread the data out when hovering over a point. This is done by using the pixel position of the mouse and inverting it using the polylinear time scale defining the current x-axis. The time data defining the domain of the polyulinear timescale is then adjusted by creating an array with the endpoints of the time data, the mouse location in the center, and time points one minute before and after the mouse location. The range of the polylinear timescale is an array with the mouse location in the center and the minute before and after shifted by 50 pixels to the left and right.

```
var mouse = d3.mouse(this);

//get time of mouse location
mousetime = polylinearTimeScale
  .invert(mouse[0]);

timeup = d3.time.minute
  .offset(mousetime, 1); // +1 min
timedown = d3.time.minute
  .offset(mousetime, -1); // -1 min

// shift time by +/- 1 min
warptimedata = [data[0].time,
  timedown,
  mousetime,
  timeup,
  data[data.length-1].time];

//shift data by 50 px
shiftscale = [shiftscale[0],
  mouse[0]-50,
  mouse[0],
  mouse[0]+50,
```

Figure 4. Highly compressed data to to fish eye distortion of the time axis. This can be hard to interpret and can misrepresent the data.

```
  shiftscale[shiftscale.length-1]];

// Update scale domains
polylinearTimeScale
  .domain(warptimedata)
  .range(shiftscale);
```

While this helps spread the data locally, the dynamic motion makes clicking points more difficult. In Figure 3-4 the data is shifted between 1:05 and 1:06. Having the mouse location stay centered keeps the point of interest from moving. However, while hovering through the data, the points shift too much to be easily selected. This breaks the ability for the data to be used as effective control points for selection. Additionally, the warped time scale completely obscures the larger patterns of the data as the scale begins to overlap data that was originally not overlapped. An example of the data patterns falling away can be seen in Figure 4 where the mouse is centered between 12:46 and 12:47 and the data before 12:46 is over compressed.

Although these alternate time scale designs provided interesting new ways to see with the data, I ultimately used the linear time scale for the final system as this provided the most accurate representation of the data. I refocused how to give the analyst focus and easy control points by exploring interactivity with the data.

Interacting with the data

Interaction with the data is done through hovering, clicking, and highlighting. All interactions were designed to help the analyst quickly focus on specific parts of the data so that they could quickly view relevant areas of the video.

Hovering

While hovering over the data, each point is brought into focus using a tool tip to give the value of that data point. For the speed data, the numerical speed in miles per hour is shown. This occurs anywhere within the area of the chart as selecting and following the line is too difficult. For the wizard speech data, the statement sent to the car by the wizard is shown when hovering over the a data point, as seen in Figure 2. Since each circle is a control point, the data is only shown when hovering over the circle itself. While the mouse is over one of the speech circles, the radius is increased to 3x the original size and the opacity is changed to 0.5. A text-box appears with the exact message sent from the wizard to the car. These features help to bring the specific data point into focus, allowing the analyst to clearly see which data point they are selecting and the value of the data. The animations of these changes are near instant so that the analyst can quickly scan through the data to select a questions they may wish to see on the video. A subtle fade out time is used on the text tool-tip to keep the text up slightly longer so that the analyst can move away and still keep reading.

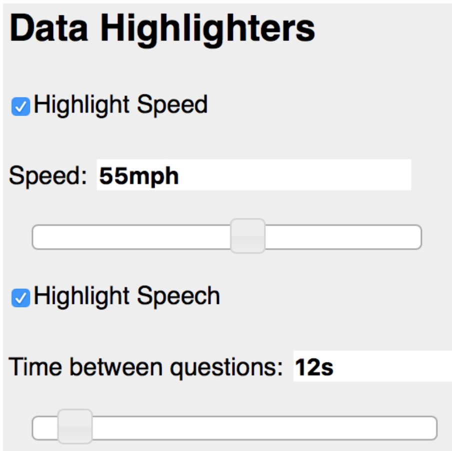

Figure 5. Controls for data highlighting. Each slider controls live highlighting of the data. The speed slider highlights speed above the selected value and the speech slider highlights areas where the wizard sent messages within the selected time.

Clicking
Each data point in the charts is clickable and linked in time with the video. When the analyst clicks on a specific data point in either the speed or speech data, the video player automatically seeks to that time in the video. Conversely, the analyst can scrub through the video using the video player playhead and the gray playhead on the charts will automatically move to show the video time in relation to the data.

Highlighting
Highlighting of the data allows for the analyst to further find interesting areas of interaction in the videos by focusing on specific areas of the data. I have implemented two simple threshold filters, 1) Speed above a certain point and 2) Minimum time between wizard text-to-speech messages. This highlighting is automatically calculated and the areas are drawn in real-time on each chart. Range sliders, as seen in Figure 5 are used to dynamically show the areas of interest, as shown in Figure 6. The highlights occur across each chart so that areas of interest in one chart can be explored in another chart. For example, highlighting speeds above 55 mph and showing this highlight in the speech chart can help an analyst to see what questions the wizard asked during these moments.

RESULTS

System Functionality & Usability
The current prototype successfully provides all the functionality described earlier and allows for fast, interactive scrubbing of the video by using the data as a guide. Analysts can click on any point within the data an seek directly to that moment in time. One can confirm that the data and video are synchronized by clicking one of the wizard speech data points to instantly hear the associated message when the video seeks to that moment in time. Seeking to specific moments in the video is near instant, however some delay can be caused by video buffering. The gray playheads in the data update smoothly during normal video play and jump immediately when the video resumes playing after the analyst seeks to a point in the

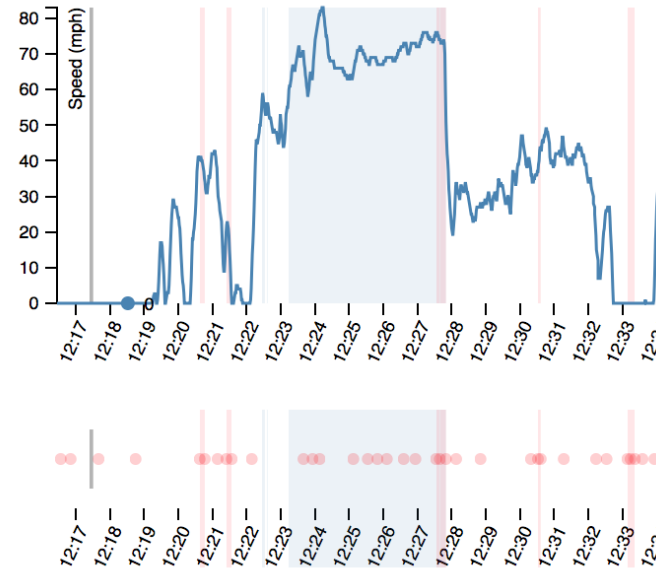

Figure 6. Highlighted data. Blue areas represent speed above 55mph and red areas represent areas where wizard speech event occurred less than 12s apart.

video using the video player playhead or clicks on a data point. Using a stable high speed internet connection limits this delay.

Animations of the data, including live tool-tips and data highlighting, are very interactive ($> 0.1s$ refresh). This provides smooth interaction with the tool. The data highlights update as the analyst slides the range slider, providing instant feedback about the what the sliders highlight.

The visually synchronized linear time scales show the data accurately with relation to time and to each of the data streams. One can quickly work across data streams to understand interesting areas of the video to explore and avoid areas of non-interest. Highlighting across both data streams allows one to quickly make comparisons across the data streams and to seek to those areas in the video.

User Interactions
Although I have not conducted a rigorous usage study of the tool, I have compiled some brief notes about how people have responded to and interacted with the system. These people include members of my lab who use similar tools in their work and visitors of a live demonstration.

Time scales
Most people found the linear time scale to be the easiest to use. People often appreciated the semi-transparent circles used in the 1-D data display as the overlapping data provided a natural highlight of potential areas of interest in the video. They also commented that this scaling gave them a good understanding of the general patterns of the data.

People found the equally spaced data to be easy to click, but confusing to understand. Although they understood the inverse relationship of the time scale with the data density, they could not readily understand patterns in the data.

The backspaced smoothing time scale was met with mixed reviews. Some people found the data to be harder to interpret due to the distorted time scale, however others found that it did accomplish the goal or reducing the whitespace while preserving the general patterns of the data. Still, most people found that the simple, linear time scale preferable and more accurate.

The dynamic fish-eye time scale was generally not preferred. People found that while the distortion did help to focus on areas of the data, especially areas with heavy occlusion, the usability was not conducive to exploring the data. Part of the issue may be my specific implementation of the filter. When testing an early implementation that centered the scale on the point above the mouse, I found that it was very difficult to click on a specific point as the target kept moving as the mouse moved. This made using the data as a control surface very difficult and aggravating. This was issue was improved somewhat by centering the fish-eye distortion around the mouse location and only distorting the view when a point was selected. Still, after the initial novelty of the data moving waned, people found little utility in having the data automatically spread out. The compression of data in some parts to the chart caused them to forget the patterns that were in the data before. Also, spreading out the data around the mouse caused the pattern of that data to also be lost. With the data spread out, analysts could easily forget how dense the data was before. Overall, people found that dynamically distorting the view was not useful for efficient analysis. Instead, people suggested that simply focusing on the point through size or color change could achieve the goal of giving an analyst focus and context in the data view. This advice was taken into account for the final prototype implementation.

Data Highlighting

The data highlighting tools were not instantly apparent to people upon first using the system. People often asked what the vertical rectangle on the data signified. However, upon either turning a filter on/off or moving the range slider, the function of the highlighter became apparent. The very fast updates of the highlighters helped people to understands what the filter was doing. Due to the speed of the updating, users often reacted to the highlighting as one of the more delightful aspects of the system.

Data features guiding video exploration

After spending time with the system, people often were able to formulate some questions about the interaction based on the data. For example, just before 12:26 (as seen in Figure 6), there is a very fast deceleration from 70 mph to 20 mph. People often asked question such as "What was the driver doing?" or "What was the wizard asking?" During these moments they used the data to seek to that point in the video, where they could see the driver beginning to exit the freeway. People also found areas where the different data highlights overlapped to be interesting areas to explore the video. People also often thought of other highlights that could be used, such as peak detection or speech from the driver. For example, one person found that the peaks seemed to correlate with the driver passing another car. These examples suggest that the tool does help people to use the data as a means to explore the interactions in the video.

DISCUSSION

The current functional prototype of the video interaction analysis tool explores a number of areas for supporting interaction researchers in analyzing video with time synchronized data. Below are three themes that emerged over the design development and brief testing of the tool.

Data as a map

The tool builds heavily on the work of VCode and ChronoViz to enable exploration of video, and extends these tools by focusing on using the data as the primary means to interact with the video. Rather than watching the video in a linear fashion, the data provides a way to rapidly explore the video non-linearly. This allows for the analysis process to be more question driven rather than time driven. People use the data patterns to explore the interactions in the video quickly and with direct purpose. Having the associated data also allows for one to begin determining correlations or causalities between the data patterns and the behavior of the driver. This change from a linear approach to video interaction analysis to a data-driven approach suggests an opportunity for the tool to make video interaction analysis less time consuming and more engaging.

Control points vs. view points

There is a difference between data that is only viewed and data that is used to control an interface. If the data is to be useful as a control point then it should minimize Fitt's Law, reducing the time required to select the correct point [11]. Although dynamic fish-eye distortion may be interesting to look at, it can make the data nearly unusable as an interactive control surface.

During the design development, I originally thought that the visual representation of the data would be more conducive to analysis. However, as I used the prototype to scrub through the video I found that the interaction was much more important than the time scaling. Overall, the presentation of the data should be kept simple and not get in the way of how the analyst interacts with the data. In many ways this echoes Bill Moggridge describing how after designing the form of the first laptop he realized that how the person interacted with the software was much more important [7]. Although I originally began by wanting to explore how visually distorted time scales may help the analyst, my design explorations pushed me more toward using interactive elements in order to make the visualization into a control surface for exploring the video.

Linked highlighting for exploration

Using interactive elements to provide focus to the data allowed for people to more rapid generate new questions about the data. The data highlighters help an analyst to quickly find potentially interesting ares of the video. This works through highlighting the areas of interests, such as speeds above 55 mph, but can also work by not highlighting the areas of low interaction.

For example, when the wizard speech slider is pulled all the way to 120 second between questions we see two areas emerge

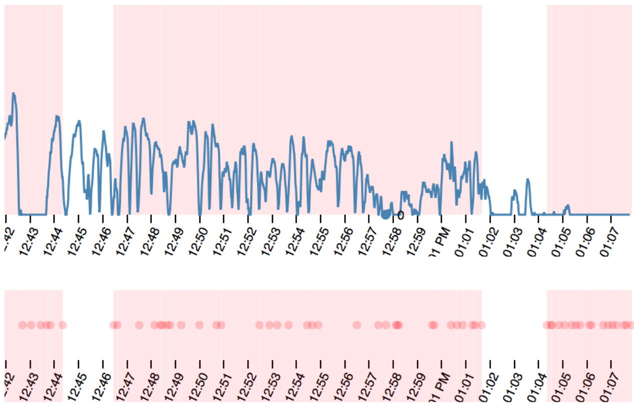

Figure 7. Areas of low speech interaction are not highlighted. This allowed for exploration of areas that may have otherwise gone overlooked.

in the data where no questions were asked, as shown in Figure 7. While this may be overlooked as merely whitespace without the highlighting, the highlights make these two moments stand out. In these particular sequences, the driver was very focused, looking for parking. It is possible that the wizard was allowing the driver to focus. Interestingly, right after the second parking attempt, the wizard begins to help the driver find parking and speaks many times within 3 minutes.

Overall, the automatic highlighting across both data sets helps people to begin formulating their own questions about the data. The fast interactivity seems to invite them to quickly explore why certain patterns occur. This helps them quickly move to those parts of the video and see what was happening on the road.

FUTURE WORK

Currently, the system accomplishes the goal of linking the time-series data streams to the video. Moving forward, I plan to add the five additional data streams that were captured during this drive so that they can also be used to explore the data. I also plan to have the data automatically highlight as the video passes over these moment in time. For example, as the video passes of over a question, it can automatically highlight it in the data chart. Another area of future development is to further build to tool automatically load different data sets and videos.

Another area of exploration is adding more visual display of the momentary information to the view. It was suggested that the dashboard of the car could be recreated and updated dynamically underneath the video in order to give the analyst a sense of what the driver can see in the car. This live updated numeric data may enhance how the video is viewed, giving the analyst a better sense of the rate of change of certain data. For example, seeing the speed drop quickly may prompt the analyst to explore the data charts and see what else was changing at the same time. This time-dependent display may help to reveal patterns that might otherwise go overlooked in the static visualizations.

CONCLUSION

In this paper I have presented a system for using time-synced data as a map to explore video data. As everyday products become more imbued with sensors, computation, and connectivity, there will be many opportunities to use data as a means for understating human experience. However, while data can be quite revealing, it will still be difficult to understand the more qualitative experiences and behaviors of people through data alone. By using quantitative data as a guide for exploring qualitative interactions in video, we can leverage computational and human abilities to generate and make sense of human experience.

ACKNOWLEDGMENTS

I would like to acknowledge Hamish Tennent, David Sirkin, Mishel Johns, and Wendy Ju for their feedback on the design of the video interaction analysis tool. I would also like to thank the CS448-b teaching team for their guidance and feedback.

REFERENCES

1. Maneesh Agrawala and Chris Stolte. 2001. Rendering Effective Route Maps: Improving Usability Through Generalization. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*. ACM, New York, NY, USA, 241–249. DOI: http://dx.doi.org/10.1145/383259.383286
2. Nils DahlbÃd’ck, Arne JÃűnsson, and Lars Ahrenberg. 1993. Wizard of Oz Studies: Why and How. In *Proceedings of the 1st International Conference on Intelligent User Interfaces (IUI '93)*. ACM, New York, NY, USA, 193–200. DOI: http://dx.doi.org/10.1145/169891.169968
3. Adam Fouse, Nadir Weibel, Edwin Hutchins, and James D. Hollan. 2011. ChronoViz: A System for Supporting Navigation of Time-coded Data. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems (CHI EA '11)*. ACM, New York, NY, USA, 299–304. DOI: http://dx.doi.org/10.1145/1979742.1979706
4. George. W. Furnas. 1986. Generalized Fisheye Views. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '86)*. ACM, New York, NY, USA, 16–23. DOI: http://dx.doi.org/10.1145/22627.22342
5. George W. Furnas. 2006. A Fisheye Follow-up: Further Reflections on Focus + Context. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*. ACM, New York, NY, USA, 999–1008. DOI: http://dx.doi.org/10.1145/1124772.1124921
6. Joey Hagedorn, Joshua Hailpern, and Karrie G. Karahalios. 2008. VCode and VData: Illustrating a New Framework for Supporting the Video Annotation Workflow. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '08)*. ACM, New York, NY, USA, 317–321. DOI: http://dx.doi.org/10.1145/1385569.1385622

7. Gary Hustwit. 2009. Objectified [Documentary]. *United States of America.(Available from Gary Hustwit, http://www.hustwit.com/category/objectified)* (2009).

8. Brigitte Jordan and Austin Henderson. 1995. Interaction Analysis: Foundations and Practice. *The Journal of the Learning Sciences* 4, 1 (1995), 39–103. http://www.jstor.org/stable/1466849

9. R. Kincaid. 2010. SignalLens: Focus+Context Applied to Electronic Time Series. *IEEE Transactions on Visualization and Computer Graphics* 16, 6 (Nov. 2010), 900–907. DOI:http://dx.doi.org/10.1109/TVCG.2010.193

10. M. Krstajic, E. Bertini, and D. Keim. 2011. CloudLines: Compact Display of Event Episodes in Multiple Time-Series. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (Dec. 2011), 2432–2439. DOI: http://dx.doi.org/10.1109/TVCG.2011.179

11. I. Scott MacKenzie. 1992. Fitts' Law As a Research and Design Tool in Human-computer Interaction. *Hum.-Comput. Interact.* 7, 1 (March 1992), 91–139. DOI: http://dx.doi.org/10.1207/s15327051hci0701_3