Exploring Video Streams using Slit-Tear Visualizations

Anthony Tang[1], Saul Greenberg[2], Sidney Fels[1]

[1]Human Communication Technologies Laboratory
University of British Columbia
Vancouver, BC, Canada V6T 1ZN
+1 604 822 4583
{tonyt, ssfels}@ece.ubc.ca

[2]Department of Computer Science
University of Calgary
Calgary, Alberta, Canada T2N 1N4
+1 403 220 6087
saul.greenberg@ucalgary.ca

ABSTRACT
Video slicing—a variant of slit scanning in photography—extracts a scan line from a video frame and successively adds that line to a composite image over time. The composite image becomes a time line, where its visual patterns reflect changes in a particular area of the video stream. We extend this idea of video slicing by allowing users to draw marks anywhere on the source video to capture areas of interest. These marks, which we call *slit-tears*, are used in place of a scan line, and the resulting composite timeline image provides a much richer visualization of the video data. Depending on how tears are placed, they can accentuate motion, small changes, directional movement, and relational patterns.

Categories and Subject Descriptors
H.5.2 [**Information Interfaces and Presentation (e.g., HCI)**]: User Interfaces.

General Terms. Algorithms, Human Factors.

Keywords
Information visualization, video analysis, video history, timelines.

1. INTRODUCTION
Digital video cameras are now commonplace and used in numerous settings. We use them to film events, as in a home or commercial video. We use them to transmit real time video for others to look at, as in teleconferencing. We also use cameras to record a fixed scene for later review, as in security or data capture situations. We define this last type of video as *stationary video scenes,* where the footage is typically captured by a strategically positioned fixed-mount camera. Such scenes typically comprise a fixed background and one or more objects that change or move within that scene, e.g., people moving in and out of view, the operations of machines, and so on. We are particularly interested in supporting *the rapid examination of a video record for events and patterns of interest in a stationary video scene*.

The naive way to review video is to simply replay it, but this can be time consuming and tedious. A somewhat better way is to use *video scrubbing* (as typically found in non-linear video editors), where moving the cursor over the video timeline plays the underlying footage at a speed corresponding to the motion of the mouse. This allows one to rapidly review video footage, and to replay segments at slower speeds (e.g., by slowly scrubbing back and forth). While better than replay, scrubbing can still be tedious for long videos. It is also error-prone. Events are easily missed if they are brief or of relatively small in size in the video scene, especially if one's attention wanders or if one scrubs too quickly.

Tang, S., Greenberg, S. and Fels, S. (2007) Exploring Video Streams using Slit-Tear Visualizations. Report 2007-886-38, Dept. Computer Science, University of Calgary, Calgary, Alberta, Canada. T2N 1N4. December.

Automated techniques can help accentuate footage of interest. With *keyframe previews*, a single frame is selected from video scenes in a way that typifies that sequence, e.g., one that is the most visually similar to all others [11]. Multiple keyframes representing a longer sequence can be presented as a storyboard or slide show [6]. With *change detection*, video footage is analyzed and marked if it deviates from the background scene or from surrounding frames (e.g., as found in many video surveillance systems such as D-Link's securicam's surveillance software www.dlink.com). Such automated methods are at their best for drawing a person's attention to single points in time, or for marking events of potential interest. Yet they do not reveal patterns as they appear in the video through spans of time and space. For example, consider a camera capturing traffic through an intersection. The above methods may show when a car passes through that intersection, but they will not easily reveal how many and at what rate they pass through that intersection during rush hour compared to a non-peak hour.

Consequently, we designed a real-time interactive application that supports the rapid exploration of a video record—not only to find events of interest, but also to see patterns between those events. To foreshadow what is to come, our method allows a person to draw marks, which we call *slit-tears*, anywhere on the source video to represent areas of interest. For every frame, the pixels under these marks are concatenated to a composite image, thereby creating a rich visual timeline of the selected video data. Depending on how tears are placed, the visualization can reveal things like object motion, small changes, directional movement, and relational patterns in the video scene.

We first detail how slit-tears builds upon previous methods of slit scanning and video slicing, and then explain our algorithm and interface. We then show by example how this new visualization allows us to create and inspect scene visualizations at two analytic levels:

- *event level,* where change events—even if temporally brief or spatially "small" —can be made more salient;
- *pattern analysis level,* where periodicity is revealed, patterns can be compared and correlated over time, and directionality and velocity of movement can be gleaned.

2. RELATED WORK

After briefly showing how videos are displayed in traditional timelines, we summarize *slit scanning* as done in photography. We then show several different approaches for visualizing video data as a timeline that have evolved from slit scanning.

Traditional timelines. Most non-linear editors offer a visual timeline as the primary way for editors to view and compose video sequences; Figure 1 gives an example. Clips in the timeline are identified by a thumbnail of the first frame in the clip. To see clip details, a person scrubs over the video by moving the timeline control. As mentioned, such timelines are largely unsuitable for analyzing video data for small events (in time or space) or for identifying patterns.

Slit scanning in photography passes a panoramic film strip rapidly across a single vertical slit, which exposes film to only a narrow slit from a scene (indeed this is how most digital scanners work). Objects moving in a stationary scene *over time* are seen as motion captured *over space* [2]. Consider Figure 2a, where the slit was positioned over the finish line. The horses are captured over time as they move across the slit / the finish line, allowing for easy and accurate judging. Similarly, Figure 3b illustrates the use of slit scanning to capture the motion of a hammer thrower [10].

Slit scanning in video, sometimes known as video slicing, achieves a similar effect. Here, a slit is placed over a video frame, and the pixels of successive video frames under the slit are captured and added to a composite image. Levin [7] catalogues many examples of slit scanning in video. Figure 3 illustrates our own TimeLine [8] system, designed to facilitate temporal awareness of a remote collaborator. The viewer positions the slit atop the live video frame by moving the red vertical line, and the views in the four rows are updated dynamically to reveal what has happened over different time [1]. The top row is the last minute (we see he has not moved much). The 2nd row is the last hour (he arrived about 20 minutes ago, although people have passed in front of the slit briefly before that). The 3rd and 4th row show the last day and the last week (we now see rhythmic pagers over daytime and nighttime). What sets TimeLine apart from other slit scanning video systems is that it provides users with a dynamic interface to the video slicing mechanism. People can interactively change the location of the slit, immediately updating the entire visualization. People can also scrub the underlying video if they see a pattern of interest. Finally, they can retrieve detailed video of events in the distant past by selecting a past point in time. For example, if a frame is selected from several days back, the day, hour, and minute view are all updated to show the surrounding frames at fine-grained resolution. Taken together, these features make TimeLine effective in allowing users to actively explore temporal patterns of behavior visualized in a 2D timeline.

Video volumes. We can also consider a video clip as a volume, where successive video frames are stacked atop one another, i.e., as a video cube [3]. A visualization of video data over time is then created by slicing of the volume with a plane or other geometric shapes [3][5]. Figure 4 gives an example, where a plane captures a diagonal view of the video cube, thereby crossing both time and space. Slit-scanning can be viewed a subset of this method, as it realizes the specific case of an intersecting plane being placed perpendicular to the face of the cube. Video volumes are more general, as different effects can be achieved by using other geometric slicing shapes and other slicing positions.

Figure 1. Adobe Premiere Pro CS3 Timeline

Figure 2. Slit-scanning. (a) judging photo finishes, from[3] (b) George Silk Hammer Thrower, from [10].

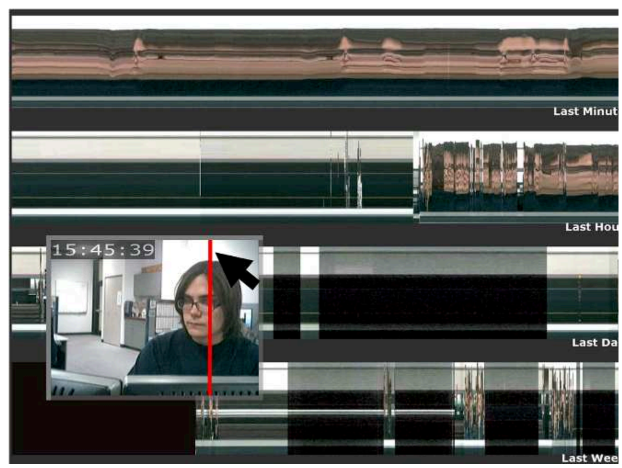

Figure 3. TimeLine, showing several days of webcam capture.

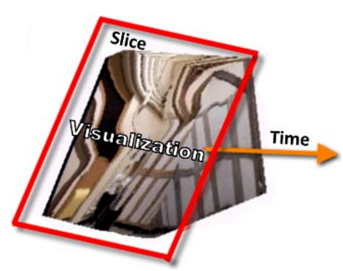

Figure 4. A sliced video cube, modified from [5]

While the non-traditional timelines above all produce somewhat abstract visualizations, in practice, people quickly learn how to read them. For example, Nunes et. al. report that people can quickly discover events of interest and patterns of activity in a telecommuter setting—to the point that significant privacy concerns are raised [8]. Yet slit scanned video is still used mostly for artistic purposes, e.g., to create interesting images or unusual film effects, or as part of an interactive art installation [7].

We believe that slit scan video techniques can be extended and practically applied to video data analysis. In particular, the following section introduces the idea of *slit-tears,* and later sections will show how it can:

- support post-hoc analytic exploration of video data,
- visualize spatial and temporal patterns,
- draw attention to brief or spatially small events,
- accentuate motion,
- indicate directional movement, and
- show relational patterns.

3. Slit-tears

The method. We have already seen how slit scan photography (and later slit scan video) captures a linear—usually vertical—slice of a frame's area, and portrays these as instances in time in a visualization. Yet, while photography is technically limited to a single linear slit, there is no need to impose this arbitrary limitation to digital video. Instead, we can capture a moment in time as multiple *slit-tears* in the video scene that are concatenated together to form a single column in the visualization.

With tear-based video slicing, we first allow the end user to draw multiple slits atop a frame's surface. Each slit is an arbitrary stroke—a straight line, curve, or scribble. For each frame, the system then captures the pixels under each slit in the order and direction that each stroke was drawn, and aligns these pixels into a single vertical column. It then appends this pixel column to the right side of the visualization. Because stroke order and stroke direction dictates how each column is drawn, a person can arrange several slits close together (e.g., for seeing relationships; this will be discussed in subsequent sections).

To illustrate this, consider the sketch in Figure 5. The bottom row represents several video frames that capture the top-down view of a room with three blue doorways. The top row is the visualization created from a series of slit-tear columns, with each column capturing the next frame in the series. A person has drawn 4 strokes in the order numbered in the left-most frame. Let's now consider how the visualization captures particular events over time by comparing several frames with their corresponding slit-tear columns.

We see a person walking out of the right door across slit 2 (columns 2-4, frame 3), and then moving in front of the bottom door across slit 4 (columns 6-11, frame 7). While this is happening, a 2nd person walks out of the left door across slit 1 (columns 7-8, frame 7), and then also stands in front of the bottom door on slit 4 (columns 9-11, frame 10) until finally leaving out of the right door across slit 2 (column 12, frame 12). The two are also seen standing side by side (columns 9-11, frame 10). While abstract, a viewer can learn to read this visualization to interpret the flow of traffic in the room, infer casual encounters that take place as people move through that room, and optionally scrub over particular areas of interest to verify what actually happened.

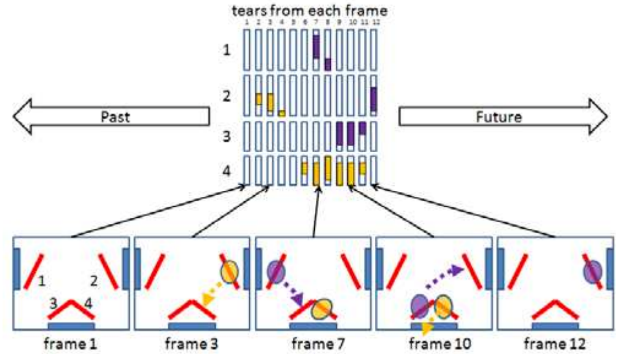

Figure 5. How slit-tears create a visualization

The system. We have created two systems that realize slit-tears in video-slicing; their combined capabilities are described below. Figure 6 illustrates a situation similar to Figure 5. The left side is the video player, while the right side shows the visualization. The video player normally shows the current frame from the running video. However, if a person scrubs over the visualization, a red scrub line appears (right side) and the video player immediately displays the frames from that scrubbed moment in time.

People create one or more slit-tears atop the live video frame, using the line or sketch tool (Figure 6, bottom left). In the figure, the analyzer began by sketching tear 1 atop the frame to capture traffic going through the left doorway. Similarly, tear 2 captures the right doorway, and tears 3 and 4 capture movement and activity around the opening to a seating area. All lines were drawn from the top down. This generates the visualization, where we see four horizontal regions representing the frame pixels underneath these lines stacked atop each other (in this image, each region is separated by black, and numbered to show the slit-tear line it represents). As in Figure 5, moving left to right in the

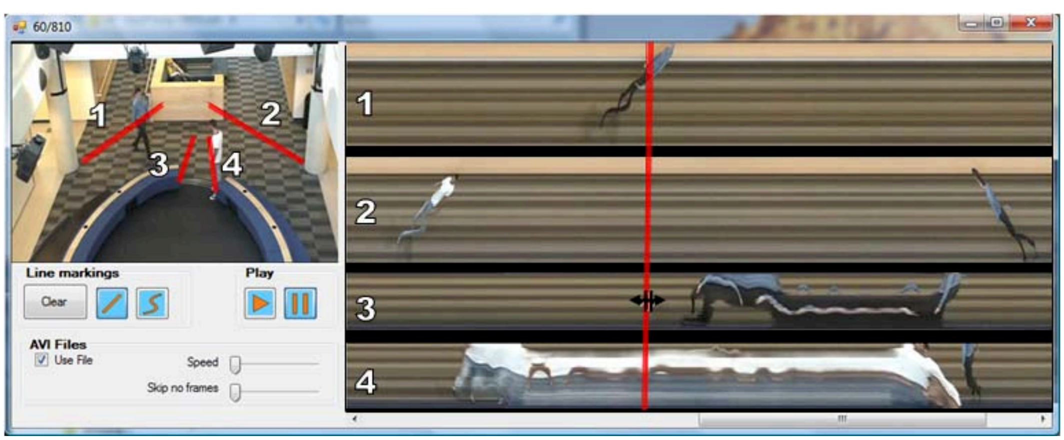

Figure 6. Screen snapshot of our slit-tear video slicing system.

Figure 7. Monitoring pedestrians entering or passing by a store in an outdoor mall. Annotations show door and people events.

timeline visualization, we see one person wearing a white shirt and blue pants has quickly walked through doorway 2 (2^{nd} row), and then pauses by the opening (4^{th} row). Shortly afterwards, the 2^{nd} person wearing a blue shirt and black pants comes out of the left door (1^{st} row), lingers at the opening along with the 1^{st} person (3^{rd} and 4^{th} rows), and then leaves through the right door (2^{nd} row). The first person disappears by continuing through the opening (the blur as he leaves suggests his direction of movement.

The system works with both live and previously-captured video (AVI files). For live video, the visualization updates itself with incoming video frames, generating new columns corresponding to the slit-tears. For previously-captured video, there are two options. One can treat it like live video, where tears and updates for subsequent parts of the visualization are done as the video plays. Alternately, one can update the entire visualization—past, present and future frames—to reflect the slit-tears.

For previously captured video, the person can also select different playback speeds (the speed slider in Figure 6) and the level of granularity of playback (the skip frames slider). When frames are skipped, details may be lost. However, the visualization then gives the viewer a broader picture of when and where events happened over longer periods of time.

We have now defined the slit-tear method and illustrated our system. Next, we will show how slit-tears can be used for user-driven analyses of video data, where strategic placement of slit-tears can create engaging and useful visualizations. In particular, we will show an analyst can use tear-based video-slicing to explore video data at both an event-level (incidents that occur at a point in time), and a pattern-level (incidents that repeat over time or over space).

4. EVENT-LEVEL ANALYSIS

Slit-tears can help visualize and emphasize events of interest. Consider Figure 7. The top left corner shows a cropped region from a low-quality 320 x 240 video. The scene is an outdoor mall, with a sliding door into a particular store at its center. The video analyzer is interested in what happens in a small, somewhat blurry region within this video: the traffic patterns around the door. She draws a single horizontal line across the typical path (labeled 1-path), and then scribbles a line over the doorway (2-door). The resulting visualization (top right) shows an interval in time. We will use this visualization to explain how the analyzer can ask and answer particular questions about events in this segment.

How often do people walk by the entrance? The visualization contains solid uninterrupted horizontal lines when the scene is static (e.g., far left and far right regions). Perturbations in these lines happen when people walk by. To answer this question, she simply compares the ratio of static *vs*. perturbed regions in the video.

How many people walked by? Individual people are seen as streaks in the top of the visualization. In this case, we see that three streaks—three people—were captured in this video segment.

Which direction were people coming from? In Figure 7, the streaks all diagonal downwards over time. This means that all three people are moving from left to right. The reason is that these people have entered line 1 at its left side, which displays those pixels at the top of the slit column. As they move through the line, their captured pixels appear lower in the column. If a person was walking left to right, the streak would diagonal upwards.

How many people entered the store? The opening and closing of the sliding door is clearly seen as a blackish perturbation on the mid-left side of the visualization's lower half. We also see that this was caused by P1: his streak at the top slows down and then disappears abruptly as he enters the door, and his colors 'dissolve' into the open doorway at the bottom. We also see that P2 has not entered the door: his streak continues beyond the doorway at the top, and he has clearly not walked into the door at the bottom. As with P2, P3 has walked past the door: his streak is continuous at the top, and the sliding door had not opened during this interval. Scrubbing over the scene verifies this: the bottom left frame shows P1 entering the door and P2 just behind. The middle and third frame shows P2 and P3 after they have just passed the doorway.

The video in Figure 7 is of poor quality, and the analyzer is interested in only a small somewhat blurry and poor contrast region of that video. Yet the resulting visualization reveals how slit-tear video-slicing can make even obscure events highly salient to an analyst, as described below.

More generally, events of interest can be problematic to see in conventional replay of video when the image quality may be poor, events may be very brief or spatially small, and patterns over time may be hard to detect. Slit-tears are a technique that helps to overcome these difficulties.

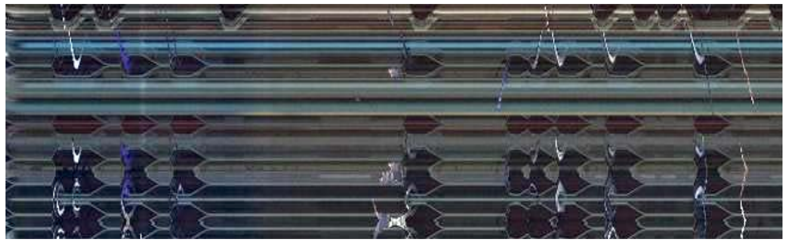
Figure 8. Several minutes of the same scene shown in Figure 8

Events are readily seen even in low-fidelity video. Slit-tears reveals change as it occurs over a region in a static background. Change highlighting works even if images are blurry, pixelated and/or low-contrast. It will also work over noisy video, as motion tends to produce regular vs. random patterns.

Spatially "small" and/or poor fidelity events can be exaggerated. Some events of interest may be spatially 'small', affecting only a modest number of pixels in the scene. These can be easy to miss or to decipher when replaying a long video. Even if we are expecting an event over just a few pixels, we can make that event highly salient simply by creating several slit-tears over that area; this enlarges the event's appearance in each column. To illustrate, we remind the reader that Figure 7 (top left) is cropped (about 1/6 of the area) from a poor quality source video. The events of interest are even smaller – each person is approximately 20x8 pixels in size, with indistinct edges. Similarly, the doorway is not much larger. Yet activity around the doorway is made salient by scribbling a slit-tear over it—the slit-tear 'expands' the doorway area to cover more than half of the visualization's height (along the bottom).

Brief events are made extremely salient by virtue of how the timeline is constructed. Tears can be conceived of as *portals of interest* in the video. In a long video scene, the timeline is a series of fairly unbroken horizontal lines; however, when objects *do* pass through the tears (or "portal"), they appear as an *intrusion* in the timeline. For example, a quick visual scan of Figures 6 and 7 allows us to rapidly spot these intrusions. As another example, Figure 8 shows the same the same timeline as Figure 7, except it is visually compressed to show over 7200 frames (several minutes of video). Even so, it clearly shows regions of interest as people move through the path and through the doorway.

5. PATTERN-LEVEL ANALYSIS

One of the key strengths of slit-tear visualization is that it allows us to easily see not only individual events, but *patterns* in the scene. By patterns, we refer to the higher level interpretation of how events inter-relate to one another over time. Because the tears traverse space, and the timeline traverses time, temporal patterns of movement and behavior are visualized spatially. As well, events can be more easily correlated when the analyzer strategically places slit-tears to juxtapose them in the timeline. Examples are presented below.

What is the interplay of cars and pedestrians at an intersection? Figure 9 illustrates a scene captured from a traffic intersection. Frames t_1 to t_5 capture this scene at particular moments in time; these times are similarly marked in the visualization. The analyzer has drawn three tears. Tear 1 follows the path of cars traveling from the top to the bottom of the main road; the painted white lines marking the crosswalk and the center of the crossing road are also visible as faint horizontal white lines in the visualization. Tears 2 and 3 follow the path of the two pedestrian crosswalks. The visualization reveals interaction patterns between pedestrians and cars across this intersection. Moving from left to right in the visualization, we first see a red car moving into the scene (the angled red streak), then stopping for a while (the red horizontal streak around t_1 to t_2). We can understand this stop by other events occurring around that time. At time t_1 in tear 1, we see a bicyclist

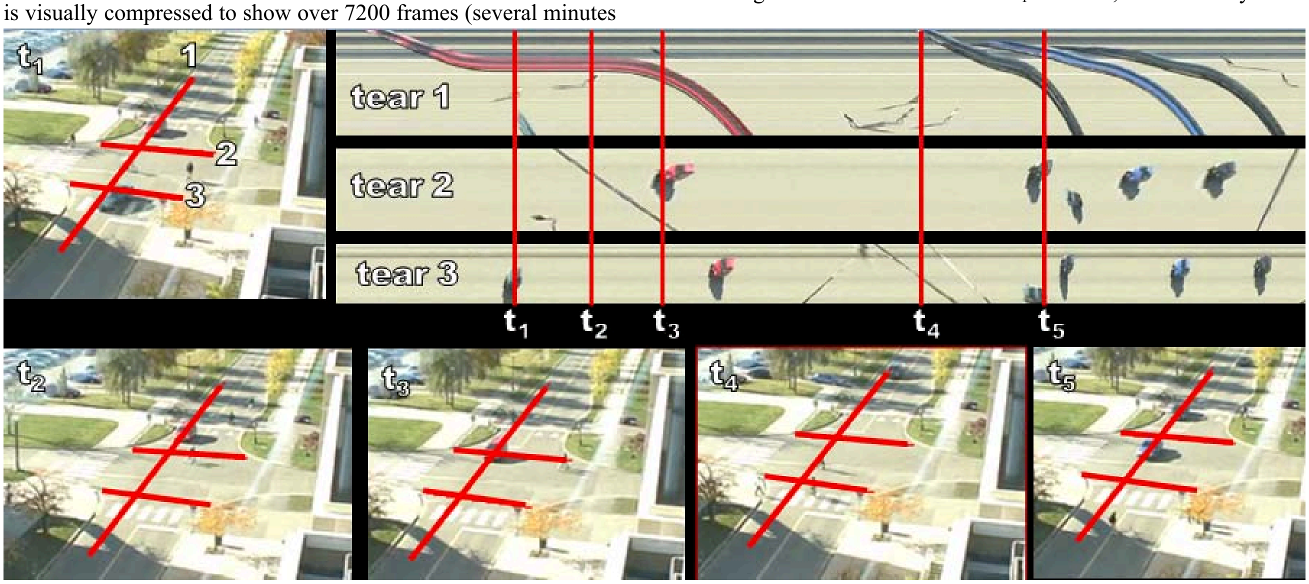

Figure 9. A traffic intersection showing the interaction patterns of cars and pedestrians

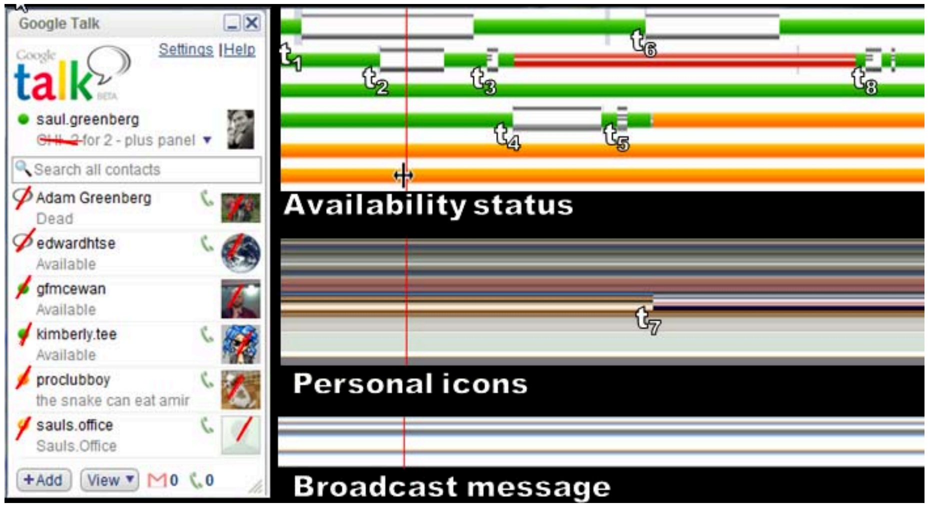

Figure 10. On-line status and talk in GoogleTalk

captured crossing the road in the center of the intersection. Looking at Tear 2 between t_1 and t_3, we see that the car is waiting for a pedestrian walking across the crosswalk (the pedestrian's path is the diagonal black streak). As the pedestrian approaches the other side of the crosswalk, the red car continues onward (we can see how far apart they are by the distance between the red car and the pedestrian in tear 2). After t_3 and until t_4, we see no cars, and several pedestrians are crossing the other crosswalk (tears 1 and 3). We can also tell their direction: those appearing as a downward diagonal streak are walking left to right, while those appearing as an upward streak are walking right to left. While this segment of the visualization does not show them, it could easily reveal other patterns, such as:

- cars that did not stop for pedestrians (e.g., a person enters the crosswalk but the car keeps on going),
- when pedestrians ran across the crosswalk in spite of approaching cars,
- near-misses.

When are people online over time and how does this lead to interaction? Figure 10 illustrates how an analyzer examines events and patterns over time in Google Talk, an Instant Messaging client. In this example, screen capture software was used to record Google Talk usage at 2 frames a second. Google Talk's interface is illustrated at a moment in time at the left. It alphabetically lists a person's contacts. Each name includes an 'availability status' icon on its left, and a personal image on its right. Conversations are shown as a white bubble, while green, orange and red icons indicate whether the contact is available, idle, or busy. The Google Talk user also gives its user the option to change a text message seen by others by typing over the field under their name (shown under the name at the top). Finally, if a person moves their cursor over a contact, the background will be shaded grey. The analyzer is currently interested in three things.

1. What is the availability status of contacts over time? She draws a slit-tear through each icon.
2. When do people change their personal image (a fairly rare event)? She draws a slit-tear through each image to capture this.
3. When does the person change their broadcast message? She draws a horizontal slit-tear through the current message so that message changes will be visible.

The timeline visualization shows several minutes of active GoogleTalk use. The grey strip preceding the conversation with the first person at t_1 suggests that the local user initiated the conversation (i.e., moving the mouse over the name, then double-clicking to initiate talk). The duration shows that it was a lengthy conversation. A parallel conversation happened at t_2 with person 2. Later conversations include person 1 at t_6, person 2 at t_3 and again at t_8, and person 3 at t_4 and t_5. A few very brief conversations (e.g., t_5) are suggestive of a quick message with the person not waiting for a response. We also see how availability status has changed: Person 2 has set their status from available to busy (the red strip), and person 4 has gone idle around t_6. Note that the local user contacted Person 2 as soon as their status changed from busy to available around t_8. We also see that one person has changed their personal icon at t_7. The constant image for the broadcast message indicates that it remained unchanged during this interval.

These examples show how tear-based video-slicing can reveal particular patterns to an analyst. These are generalized below.

Rhythms and periodicity over a tear are easy to see. Movement or events that reoccur through a tear are strikingly easy to see since they appear in the timeline as a repeated intrusion on the scene. For example, if we ran the visualization in Figure 9 for a longer time and compressed the timeline, we would likely see traffic flow over the day, e.g., peaks in the morning, lunch, and work-end, quiet times at night. Similarly, we would easily see Google Talk user rhythms wax and wane during periods of the day and even across the weekend [1].

Similar individual events can be compared as a category. When multiple similar events are occurring, the analyzer can juxtapose them in the visualization simply by the way they order their tears. For example, Figure 10 (top) juxtaposed the status of 6 people, leading to easy comparison of how conversations overlapped.

Different individual events can be correlated. Similarly, slit-tears of quite different events can be juxtaposed to see whether correlational relationships exist between objects, movement, or patterns in the video scene. For example, Figure 9 correlates vehicle movement and speed through the intersection with pedestrian traffic on the crosswalk.

Directionality and velocity can be easily ascertained. With strategic placement of slit-tears, the directionality and comparative velocity of objects moving about the scene can be easily ascertained. Figures 7 and 9 show direction by the angle of the streaks, be they people or cars. Figure 9 also shows velocity: steep angles are high speed vehicles, shallow angles are low speed, and horizontal streaks means that the vehicle is stationary. Similarly, drawing a diagonal slit-tear through a side-view of an

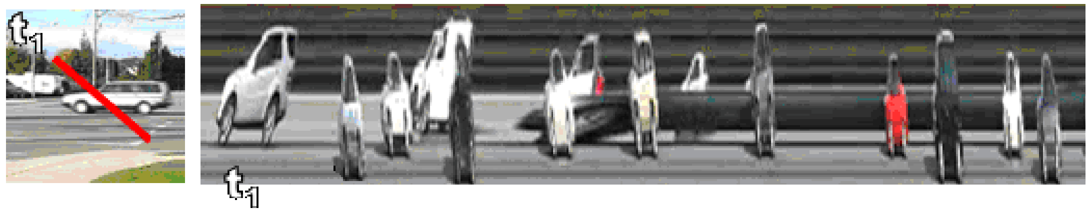

Figure 11. Car speed through an intersection

object, as done in Figure 11 (left), exaggerates direction and velocity. Cars passing through this tear are seen at different widths, and these widths suggest their comparative speed. "Longer" cars are moving more slowly than "smaller" cars; they are longer because they have stayed under the tear for more frames. For example, the extended black "limousine" in Figure 11 is actually a car waiting to make a left turn. Cars in Figure 11 are also slanted diagonally (somewhat cartoon-like), because different parts of cars going in different directions actually pass through different parts of the tear at different times. Thus, cars slanted forwards, are actually going right, while cars slanted backwards are actually going left.

6. DISCUSSION

We have introduced the concept of slit-tears and its interactive timeline visualization. We have also shown how it can be used across various video scenes in a way that reveals both events and patterns across time. We now briefly discuss some general advantages and limitations of this approach, as well as some extensions to our current system.

6.1 Advantages of Slit-tears

Slit-tears allows for exploratory video analysis. The strategic placement of slit-tears on the scene can visualize key events and patterns in the timeline. If one knows ahead of time what one is interested in, then one just places slit-tears over those areas of interest. Yet the power of the slit-tear technique is that it also allows for data exploration. The interactivity of slit-tear placement at any time (including clearing old slit-tears) means that people can use it as a tool for ongoing generation and provisional testing of hypotheses about the video data. For example, an analyzer of the traffic scene in Figure 9 could decide, after the fact, to look for jaywalking events (people on the road outside of the crosswalk), whether bicycles stop at the stop signs, the effect of traffic and pedestrians on right-hand vs. left hand turns, and so on. These and other explorations could be triggered by seeing unexpected events in the visualization.

Slit-tears is a generalizable video-analysis technique. For particular situations mentioned above, we could easily conceive of other analysis tools that might be better than slit-tears. For example, maybe other visualization methods could reveal patterns and events with greater clarity. Automated methods could analyze the scene (e.g., object recognition combined with optical flow to detect movement across frames), which could ease the analyst's burden. Alternately, we could deploy sensors at key locations to track specific types of events, thereby producing data more amenable for automated analysis, e.g., descriptive statistics.

The problem is that these alternate approaches are not generally applicable to the many every-day situations we may want to analyze. Automated analytic tools, for example, are typically only good for detecting particular kinds of events. They are also highly error-prone (e.g., as in the low-fidelity video in Figure 6 where the area of interest is only a few pixels in size). They easily miss events of interest: most operate through signal processing, which is divorced from the semantics of the objects and actors in the video itself. Sensors require heavy investment in time and materials in terms of their placement, to the point where their use is infeasible and/or illegal. Specialized visualizations need to be programmed to reveal particular kinds of information of interest. Aside from their inaccuracy and expense, these methods typically require that the analyzer knows *a priori* the events of interest, where they can then choose their tool accordingly.

In contrast, slit-tear visualization is a general approach for video collection and analysis that can augment other approaches. It requires no specialized equipment aside from an off-the shelf digital video camera, a tripod, and a computer for generating the visualization. It can be used in a broad variety of settings. This includes video captured for other purposes, e.g., surveillance and traffic cameras. It relies on no special analysis algorithms. It is an interactive visualization that lets the analyst decide upon what events should be captured by where slit-tears are positioned in the scene, and how they are interpreted.

Slit-tears is grounded in the actual data. Many statistical or analytic techniques abstract events and patterns in a scene, and present it in summary form: numbers, summary and correlation statistics, graphs, and so on. While useful, the analyzer may consequently overlook nuances of the captured data, simply because the raw data is stripped away or not linked to the summary view.

In contrast, our slit-tear tool conceptualizes the analysis space in a readily understood "camera" and "pixel" space. The timeline simply takes selected areas of the video and translates the time dimension to a spatial dimension. The raw data is still visible in this representation. Further, rapidly scrubbing over areas of interest in the timeline reveals the source video, so that details that generated the visualization can be readily examined. This not only adds to a person's understanding of the pixel renderings in the timeline view, but shows other events in the scene that may reveal why particular events or patterns occurred.

6.2 Limitations

It only works with stationary video. Our examples are all based on stationary video. Tears are most easily interpreted as "portals" into a fixed location, where changes at those locations are easily spotted in the visualization. If the camera itself moves around the scene, the visualization is much harder to decipher. Still, we do note that slit-scanning in photography has been used effectively with moving cameras; so future research could explore when such situations are appropriate.

Appropriate camera placement is crucial. The videographer should carefully consider the video camera's placement. Aside from the normal concerns of video (lighting, field of view, etc.), the videographer has to ensure that events of potential interest are captured in the field of view, and that other events will not intrude to produce spurious patterns. For example, reconsider Figure 7, where the events of interest are the pedestrian traffic around several stores. If the camera was placed (say) across the street, then cars passing along the street in front of the camera would not only block the events of interest, but generate spurious patterns in the timeline. Birds-eye view and highly oblique views tend to produce the most useful video sources.

The analyzer must be immersed in the data. The analyzer needs to take an active role in placing slit-tears, and how to interpret the visualization. In our experience, this often requires a trial and error strategy: the analyzer places slit-tears, views the visualization and explores it by scrubbing to ensure that the correct data is not only captured, but presented in a way that leads to easy analysis.

6.3 System Extensions

The actual system we developed (Figure 6) implements the basic ideas of slit-tears, but omits several features that would make it a proper video analysis tool. Several of these features were already implemented in our earlier TimeLine system [8], so their inclusion requires only routine development effort.

Overview to detailed views. For the system to be truly useful, the analyzer would need to examine the visualization across different time spans. The most detailed timeline uses a pixel column for every frame in the video (e.g., a single timeline row on a 1200 pixel-wide screen would display only 80 seconds of a 15 fps video). Yet, if a 24 hour video were to viewed in that same horizontal space, then each slice would have to represent 7.2 seconds. Clearly, events could be easily missed.

Nunes et. al. [8] addressed this problem with several strategies, as seen in Figure 4. First, the visualization shows the last minute, hour, day and week simultaneously. Second, when a slice represents more than one actual frame, the frame from that set that differs the most from the prior slice is chosen. This heuristic ensures that changes are visible in the timeline. Third, the overviews are linked: clicking on any slice generates a detailed view in the other rows in the timeline. All these approaches are possible in our system. Alternately, a fisheye view strategy could be used to visualize details in an overview [9].

Resources. Our current system keeps all video frames in memory. This is clearly impractical for very large videos. We previously suggested several resource-reduction methods [8], all applicable to our extension into tear-based video slice generation.

Editing operations. Split-tears should be individually editable, where the visualization is dynamically updated as editing is occurring (our current system only allows tears to be drawn and removed). The system should include conventional line-editing methods as found simple drawing applications. As well, the system should let a person change a slit-tear's drawing direction (to flip its appearance in the visualization), its stacking order (to reorder where it appears in the visualization, and even duplicate it (so one can strategically copy and place the same tear next to several others in the timeline).

Annotation. The analyzer should be able to annotate the video and the timeline directly. While we have simple annotation in one of our prototypes, a full annotation tool would be very handy. Ramos et. al. gives an excellent example of how annotation could be incorporated in videos and timelines [9].

7. Conclusion

Tear-based video slicing is a general and powerful user-driven video analysis and exploration technique. It is cheap to use, works with off-the-shelf equipment, and no site preparation. It can be fruitfully applied in many domains where exploration and analysis of stationary video data is required. It works in real time over live and stored video. We have illustrated several examples of its use with several application genres. We demonstrated that it is possible to study event-based occurrences in the video data, and more importantly, pattern-level occurrences.

8. Video

Static images are a poor means for illustrating a highly dynamic and interactive system. Consequently, this report includes a video, also available at http://grouplab.cpsc.ucalgary.ca/papers/

9. ACKNOWLEDGMENTS

Thanks to Michael Nunes, Carl Gutwin and Sheelagh Carpendale. Research is funded by Nectar NSERC Research Network, and the NSERC/iCORE/SMART Chairs in Interactive Technologies.

10. REFERENCES

[1] Begole, J., Tang, J. C., Smith, R. B., and Yankelovich, N. 2002. Work rhythms: analyzing visualizations of awareness histories of distributed groups. *Proc ACM CSCW,* 334-343.

[2] Davidhazy, A. Slit-scan photography. School of Photographic Arts and Sciences, Rochester Institute of Technology. Accessed Mar, 2007. URL=http://www.rit.edu/~andpph/text-slit-scan.html.

[3] Del Mar Thoroughbred Club. Accessed Nov, 2007. URL=http://www.dmtc.com/racinginfo/photofinishes/ .

[4] Elliot, E. 1993. Watch, grab, arrange, see: Thinking with motion images via streams and collages. *MS Thesis in Visual Studies*, January, MIT.

[5] Fels S., Lee, E. and Mase, K. 2000. Techniques for interactive video cubism. *Proc ACM Multimedia,* 368-370.

[6] Komlodi, A. and Marchionini, G. 1998. Key frame preview techniques for video browsing. *Proc ACM Conference on Digital Libraries (DL '98)*, 118-125

[7] Levin, G. An informal catalogue of slit-scan video artworks. Accessed Nov, 2007. URL=http://www.flong.com/writings/lists/list_slit_scan.htm.

[8] Nunes, M., Greenberg, S., Carpendale, S. and Gutwin, C. 2007.What did I miss? Visualizing the past through video traces. *Proc ECSCW'07 European Conf on Computer Supported Cooperative Work*, Springer-Verlag.

[9] Ramos, G. and Balakrishnan, R. 2003. Fluid interaction techniques for the control and annotation of digital video. *Proc ACM UIST '03*, 105-114.

[10] Silk, G. Hammer Thrower. Image reproduced from National Gallery of Australia gallery. Accessed March, 2007. URL=http://www.nga.gov.au/Silk/Gallery.htm.

[11] Yeung, M. M., and Yeo, B-L. 1997. Video visualization for compact presentation and fast browsing of pictorial content. *IEEE Transactions on Circuits and Systems for Video Technology*. 7 (5), 771-785, IEEE.