What Did I Miss? Visualizing the Past through Video Traces

Michael Nunes[1], Saul Greenberg[1], Sheelagh Carpendale[1] and
Carl Gutwin[2]
[1]University of Calgary, Calgary, AB, T2N 1N4 Canada
[2]University of Saskatchewan, Saskatoon, SL, S7N 5A9 Canada
{saul, nunes, sheelagh}@cpsc.ucalgary.ca, gutwin@cs.usask.ca

Abstract. Always-on media spaces broadcast video between collaborators to provide mutual awareness and to encourage casual interaction. This video can be easily recorded on the fly as a *video trace.* Ostensibly, people can review this video history to gain a better idea of the activities and availability of their collaborator*.* Such systems are obviously highly contentious, as they raise significant privacy concerns. However, the ease of capturing video means that video trace systems will appear in the near future.

To push the boundaries and encourage debate about video trace technologies within the CSCW community we created TIMELINE, a highly effective visualization system that combines ideas in *slit scanning* as used in interactive art to allows people to easily and rapidly explore a video history in detail. We describe its design and implementation, and begin the debate by offering preliminary reflections on how it can be used and misused. To encourage this debate, TIMELINE is freely available for others to try.

Introduction

Video media spaces (VMS) are always-on video channels that connect people and places (e.g., Bly 1993; McEwan 2005). Their primary purpose is to provide collaborators with awareness that leads to casual interaction. Once the subject of esoteric research requiring specialized equipment and networks (Bly 1993), the wide availability of inexpensive web cameras combined with the Internet and powerful home computers now let people easily create their own media spaces.

Various researchers have argued that VMS are valuable for distributed groups with a real need or desire to recreate the kinds of interactions that normally

Nunes, M., Greenberg, S., Carpendale, S. and Gutwin, C. (2007) **What Did I Miss? Visualizing the Past through Video Traces.** Report 2007-855-07, Dept. of Computer Science, University of Calgary, Calgary, Alberta, Canada, March. 21 pages.

happen when they physically work close to one another (e.g., Kraut, Egido and Galegher 1990; Whittaker, Frolich, and Daly-Jones 1994). The video acts as a surrogate for interpersonal proximity by bringing distant people closer together: one can see other people's presence and activity over the video channel. This interpersonal awareness creates opportunities for people to engage in light-weight casual interactions through the video channel at (hopefully) appropriate times and in an appropriate manner. While such always on video raises an Orwellian spectre of Big Brother, VMS are increasingly accepted by everyday computer users, e.g., when friends stay connected with one another for long periods of the day through the free digital video capabilities of some Instant Messengers, or when small communities share a collective *n*-way media space through experimental systems such as the COMMUNITY BAR (McEwan and Greenberg 2005).

Compared to computer activity indicators in Instant Messengers, video is quite good at providing rich 'at-a-glance' awareness of activities of others, which in turn lets people estimate availability for conversation. The problem is that video still misses much when compared to everyday situations where people co-habit a space (Hudson and Smith 1996). First, video demands *foreground attention*, while in everyday life people notice the activities of others in the *background periphery of attention*: for example, people see others as they walk by and their comings and goings, and they hear the sounds they make. That is, VMS only works if people are looking at their computer screen. Second, video has a narrow field of view, while in everyday life people see others as they move around the entire space. This includes how others leave things scattered around, which provides additional clues of presence (e.g., a coat on the coat rack suggests the person is in). The result is that people who use VMS do not get as good a sense of others' activities over time, which in turn affects how they can interpret what others are doing and how available they are for conversation. As a simple example, consider a glance into a media space where the distant person is not present. Unless one has been actively attending the computer, there is no way for the viewer to know if the other person has recently stepped out or if they have not yet arrived for the day. In contrast, in the everyday world one would likely have noticed the other person arrive, and would have noticed that they just left with a friend with their coffee cups in hand and without jackets: it is easy to infer that this person has gone for a coffee at the downstairs café, and will return shortly.

To partially solve the problems of current VMS, Hudson and Smith (1996) suggested that media spaces could be augmented by giving people a visualization that displays the recent patterns of activities – the *activity history* – of others. Somewhat similarly, Begole et. al. (2002) suggested that activity information can be represented as *actograms*: graphs that show rhythms in people's behaviours over time.

One way to reveal this activity history is through a *video trace* – a video history visualization that reveals important media space events over time, and that also allows easy inspection of those events. Using video is powerful, as the raw images potentially provides a more expressive record of past activity when compared to abstracted activity information (see §Related Work). Video is captured as it arrives (a simple matter with current computer technologies), and is reconstructed as an interactive visualization that reveals an overview of the video history. The idea is that people can infer relevant patterns in the visualization, and even explore it in detail to acquire a better idea of the past activities of their collaborators and what it suggests about their current availability.

Video trace systems are obviously highly contentious, as they raise significant privacy concerns. Brief embarrassing or private actions are now captured, easily found and replayed at full fidelity. Previously hidden time-based work patterns are revealed, such as how many hours someone is actually working over a day.While some would argue that we should not encourage research in this area, we strongly believe that the simplicity of the idea along with the ease of capturing video means that video trace systems will appear in the near future, e.g., created and disseminated by grass roots developers, or included in commercial systems such as Instant Messaging. The real problem is that there has been little debate about such systems. Part of this problem is that prior video trace prototypes do not provide the richness necessary to really explore their expected uses (see §Related Work). Consequently, they do not incite detailed debate about their expected uses, and the tradeoffs between awareness-gathering needs and privacy.

To push the boundaries and encourage debate about video trace technologies within the CSCW community, we created TIMELINE: a highly effective visualization system that allows people to easily and rapidly explore a video history in detail. As we will see, TIMELINE applies the overview and detail approach from information visualization to *slit scanning* as used previously in interactive arts. In the remainder of this paper, we describe TIMELINE‘s design and implementation (so others can replicate it), and how it relates to prior work. We then begin the debate by offering preliminary reflections on how it can be used and misused. To encourage further debate, TIMELINE is available for others to try at http://grouplab.cpsc.ucalgary.ca/cookbook/ (select TimeLine in the sidebar).

TimeLine

The TIMELINE visualization of a video history trace lets people do the following:

a) immediately see patterns of activity within a video history via a technique called slit-scanning;
b) use minute, hour, day and week visualizations to present longitudinal overviews views of the history at different time granularities;
c) explore patterns across different parts of the scene by moving the slit;
d) rapidly explore event details within a large video stream by scrubbing;
e) retrieve further details of the far past by selecting times of interest.

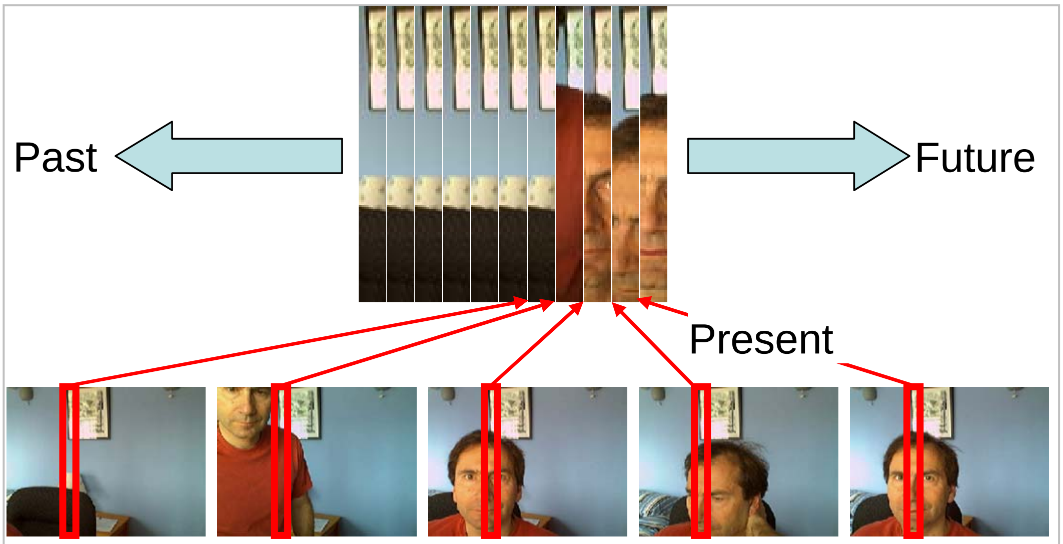


Figure 1. How video slices implement slit scanning. Slices of images are extracted from the video frames over time (bottom) and added onto the composite image (top).

This section describes these features. We stress that the static images in this paper are a poor substitute for actually using the system. We highly recommend that viewers either try our download (http://grouplab.cpsc.ucalgary.ca/cookbook/ and then select TimeLine in the sidebar) or watch our on-line companion video (Nunes, et. al., 2006). As well, we recommend viewing the paper's images in color (electronic PDF or color printing) *vs.* as a grey-scale printout.

Slit scanning

TIMELINE uses an existing technique based on *slit scanning* (Levin 2006, Davidhazy 2007) to create a composite image of video activities over time. Slit scanning, originally developed in photography (see §Related Work), exposes film to only a narrow slit from a scene; while panning the camera smoothly captures a normal scene, interesting images are created by irregular panning (i.e., spatially distorted scenes), or by introducing animated things in the scene to capture motion over space. The same approach can be realized in video by *video slicing*.

Video slicing first extracts a scan line from a video frame, and then adds that line to a composite image over time. Figure 1 illustrates this process over 5 frame sequences. To exaggerate the effect, we illustrate a slice that is several pixels wide and ~1 second intervals between sequences. The same area in each frame is extracted from the captured video frames (red boxes, bottom), and added to the right side of the composite image (top). Thus the top image portrays a pattern showing an empty room (first 7 slices), and then the person arriving in the room and sitting down behind the camera. In practice, the extracted slice is usually a single pixel thick and taken across short time intervals to give a smooth scanning effect (as in Figure 2, top) – this is somewhat similar to how normal scanners work. As well, the slice need not be vertical.

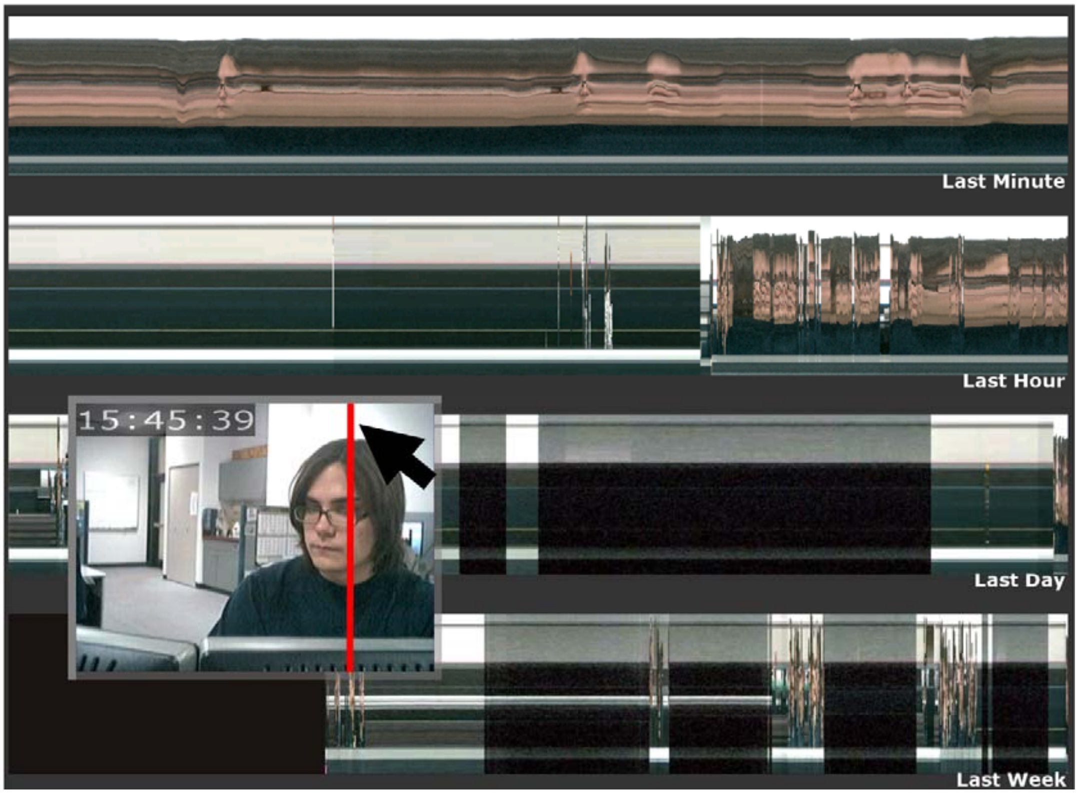


Figure 2. TIMELINE in action. The video history shows a personal workspace over several days.

TIMELINE, illustrated in Figure 2, implements video slicing using 1 pixel wide vertical slices captured at 17 frames per second (fps). The full-sized real-time video stream, shown at the bottom left of the Figure 2, is displayed within a floating window. The vertical red line within that window is the *slit focus bar*, and specifies which pixel column slice is being extracted from the video frame. The slice is added to the right side of the top row. In this case the face is 'blurred' as the person is moving back and forth across the scan line during capture.

Minute, Hour, Day and Week Visualizations

Somewhat similar to the Last Clock (Angesleva and Cooper 2005) a week-long timepiece is created by selectively adding slices at different time intervals to other rows. In this way, TIMELINE shows the last minute (1st row), hour (2nd row), day (3rd row), and week (4th row) of captured video. This process is done continuously. Thus the visualization itself, in combination with the dynamics of how new frames are added to it over time, gives viewers an overview of the short- and long-terms rhythms of activity across the composite image.

Each row captures different granularities of video frames, which means a single slice (especially in the week row) can actually represent many frames collected in a large interval of time. In these cases, and as described in §Implementation, TIMELINE uses an image differencing method that chooses the most information-rich frame to display from the many available frames.

Adjusting the Slit

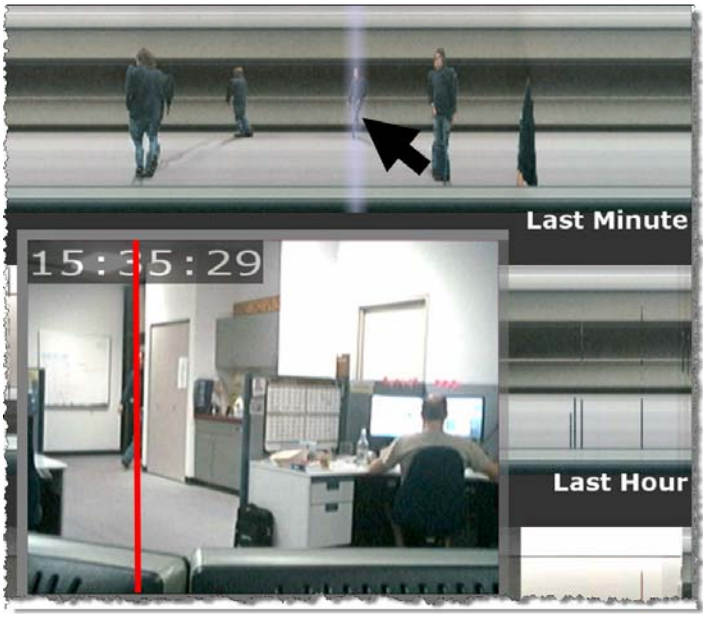


Figure 3. Moving the slit focus bar immediately updates the visualization.

One way in which TIMELINE extends on previous systems such as the Last Clock (Angesleva and Cooper 2005) is by allowing viewers to adjust the focus area of the slit. As mentioned above, the red line in the video is the slit focus bar, which indicates the column from which the video slices are taken. Viewers can interactively focus the visualization on a different area of the scene by moving the slit focus bar over a new column in the video frame. As the bar is being moved, the entire visualization is updated immediately and smoothly. Thus one can 'scan' the scene, where interesting patterns emerge within the minute, hour, day and week rows as the bar is moved. For example, Figure 3 illustrates the same overview as Figure 2. Here, the viewer has moved the slit focus bar over the doorway in order to see people as they come and go. The visualization clearly shows this activity, where slow scans of moving people are seen as they enter and leave in the minute row. The hour row shows the same activity around the doorway, except that the movement is compacted into single columns of activity. That is, they are seen as 'disturbances' in the hour TIMELINE.

Scrubbing

Another significant difference between TIMELINE and other systems is that TIMELINE allows for very rapid and detailed exploration of the video history. When a person drags the mouse to move the cursor across any of the visualization rows, the corresponding video taken at that particular point in time is rapidly displayed in the floating window. For example, quickly scrubbing over the entire day row will replay the events of that day at the same speed, i.e., a 1 second scrub replays the whole day in a second. Scrubbing is illustrated in Figure 3, where the viewer is scrubbing back and forth over an image of a person in the minute row (under the cursor and as indicated by the translucent band) to see replay details of that person entering the room in the floating window.

This scrubbing capability of TIMELINE is important, and distinguishes it as an awareness system. The constructed visualization lets people notice activities, variations and disturbances in the scene over different time scales, as represented in the minute, hour, day and week rows. Quickly scrubbing over an area of interest reveals the actual activity that produced that visual. That is, the visualization provides an overview of activity during the week, while scrubbing lets people quickly investigate details of possible interest.

Retrieving Details of the Far Past

As mentioned, quickly scrubbing over one of the visualization lines allows the viewer to replay the video at a very high speed. Scrubbing the minute view always shows all stored frames, which were captured at 17 fps. To see beyond the last minute, one must move to the hour/day/week rows. However, the granularity of the playback detail in these rows is much coarser than in the minute row, as more time elapses between selectable frames. For reasons explained in the Implementation section, each slice in the hour row actually represents about 3.3 seconds of activity. In other words, while 56 frames were seen by the system in these 3.3 seconds, only 1 is stored in the hour view. Similarly, each day slice represents 1 1/3 minutes (1344 frames), and a week slice about 9 1/3 minutes (9408 frames). Scrubbing will show the frame represented by that slice, but not the others within that interval. Yet those other frames could be important for understanding what is going on.

To remedy this, TIMELINE provides detailed exploration of the distant past by

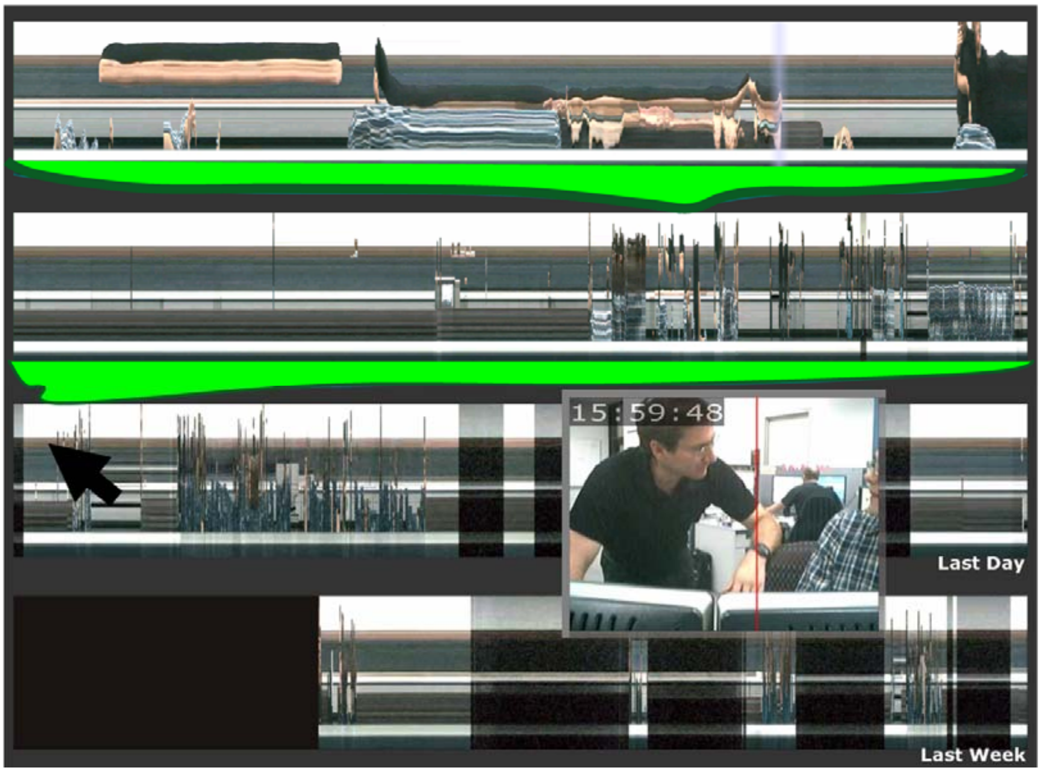


Figure 4. Regions in coarser grained rows can be selected for detailed exploration in the finer grained rows.

allowing a person to select an area of interest by right clicking in one of the coarser grained rows. TIMELINE then retrieves the detailed video around that time period, populates the more detailed rows with that video, and freezes the visualization to show how it would have appeared during that moment in time. This is illustrated in Figure 4. The person has selected a specific time of interest in the day row (cursor, far left). The detailed frames around that time period are retrieved and used to populate the hour and minute rows. Feedback to show the relationships between these time periods are indicated by the green braces between these rows. If the person right clicks, the visualization immediately returns to its normal 'live' view.

Reading the TimeLine Visualization

Previous researchers have suggested that knowing others' activities over time could help the viewer identify opportune moments to make contact with them, i.e., not only when others are reachable, but when they are likely amenable to being contacted (e.g., Begole et. al. 2002; Fogarty et. al. 2005). A key observation by Begole et. al. (2002) is that many activities often recur over days and weeks as long-term patterns, or 'work rhythms'. These patterns typically indicate: when remote colleagues arrive and depart for the day or when they break for lunch or coffee; whether they are working; how one or more visitors enter and leave the area and how long they stay; the ebb and flow of meetings and phone call frequency over the day; and differences in activities between work days, and between week days and the weekend. Fogarty et. al. (2005) further argue that details of activities are highly correlated with non-interruptibility. These include knowing things such as: whether the door is open or closed; whether occupants are seated or standing; whether they are working at a desk or table or computer; whether they are writing, using a mouse, or eating; whether they are talking on the phone and/or to guests; the number of guests present and whether these guests are sitting or standing, etc. Johnson and Greenberg (1999) also studied how people interpret images of work colleagues. They found that people are judged as less available when they are seen to be absent from their office, or in conversation with others. People are judged more available when they are in transition (e.g., entering or leaving a room), and when they do not appear to be working.

Many of these activities are hinted at in the overview visualization, and are easily seen during scrubbing (as long as they are in the camera's field of view). At first, images produced by the visualization may appear difficult to understand. Yet viewers quickly learn to read them. The full-sized frame in the floating window also serves as a good point of reference to help people understand what different patterns mean as they are being generated (indeed, our experiences are that people enjoy moving in front of the camera to create these patterns).

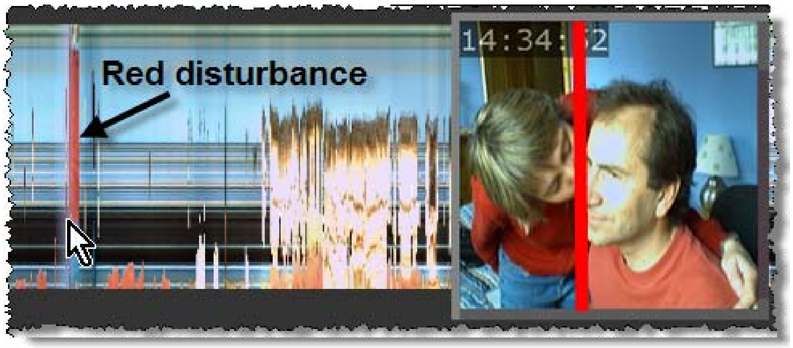


Figure 5. The visualization reveals a brief visit.

For example, consider some of the prior figures. In Figure 2, we clearly see that the person is sitting in front of their computer. The partial scans of that person’s face suggest they are moving their head only slightly, i.e., they are concentrating intently on the screen. Glancing at the hour row, we see that this person has been there for about 20 minutes. Earlier in that hour, the constant pattern suggests that the person was away. However, there are a few visual disturbances (the vertical white and grey lines), which are likely people walking by in the background. The day and week rows give a broader overview. The day row shows (from left to right), the previous afternoon, lights going off for about an hour but back on again (someone has likely left and reentered the room later), and then off again over the nighttime until the person returns in the morning. The week shows almost 5 days of day and night activity: we clearly see (going left to right) that this person has worked two partial days (the weekend) and then two full days.

As another example Figure 3 shows the TIMELINE with the focus column set on the distant doorway. As mentioned earlier, this shows people as they enter and exit the room. The visualization in the minute line shows a recognizable person as they pass through the doorway, while the hour shows these changes as single column perturbations.

Figure 4 is somewhat similar the Figure 2, except in this case we see more ‘coming and going’ activity by the person. We know it is the same person because the colors and patterns of his shirt are the same (indeed, looking at the week view we see that he has worn the same shirt for several days).

Figure 5 extracts a portion of an Hour row from a home telecommuter. Here we see a more or less regular pattern of the person working at the computer over this hour. However, a tall red line at the left of the row differs from other colors and patterns in the scene (i.e., the telecommuter is wearing an orange shirt, not a red one). This disturbance suggests that a 2nd person has briefly entered the scene. Scrubbing over that area reveals that it is the tele-commuter’s wife coming to give him a quick kiss on the cheek (as seen in the focus window).

Figure 6a+b shows how camera angles can reveal quite different information. As evident in the focus window of Figure 6a, the worker has positioned his camera to capture a side view of his desk: the telephone, keyboard and partial view of a seated person are all visible. The annotated minute view in Figure 6b reveals activity around the desk as interesting patterns. From left to right, we see an arriving person (blue jeans and shirt are visible), then fingers as he types on the keyboard. The person then picks up his phone, as indicated by the black bar (the phone pattern) changing to tan desk color and the fingers disappearing from the keyboard. We then see the phone returning and the person resume typing for a few moments. The following solid pattern suggests that the person is still there but no longer typing (because there is no 'leaving body' pattern).

Implementation

TIMELINE is developed using C# .NET and uses two toolkits: COLLABRARY and .NETWORKING (Boyle and Greenberg 2005a). The COLLABRARY provides a camera class that makes it very easy to retrieve frames from a webcam video stream attached to a particular computer. .NETWORKING is a notification server that lets people publish key/value data pairs (including multimedia data such as video frames) to a shared dictionary data structure; the system automatically takes data posted by one client, and propagates it to other clients that have subscribed to that data. TIMELINE includes a small client program that uses the COLLABRARY to capture the webcam video stream from a person's computer; it then makes these available to other clients by posting the marshalled individual video frames to the .NETWORKING shared dictionary. People wishing to view this information run the main TIMELINE client on their machines. That client connects to the shared dictionary and subscribes to the video data, The client retrieves and un-marshals that data as it arrives, and processes it as described below.

Because we wanted TIMELINE to be extremely responsive, we decided that the TIMELINE viewer should store in memory all relevant video frames in an uncompressed format. This would make it possible to instantly regenerate the visualization as people moved the slit focus bar (by quickly extracting the relevant pixel slices from all frames), or to immediately display the appropriate frame in the floating window while scrubbing. Yet it is infeasible to store in memory a single monolithic history of all frames generated over a week, as the memory requirements are far too high. Instead, the TIMELINE

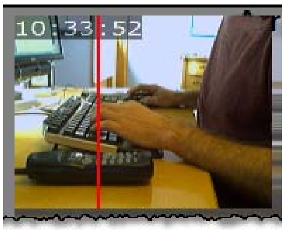


Figure 6a. The scene

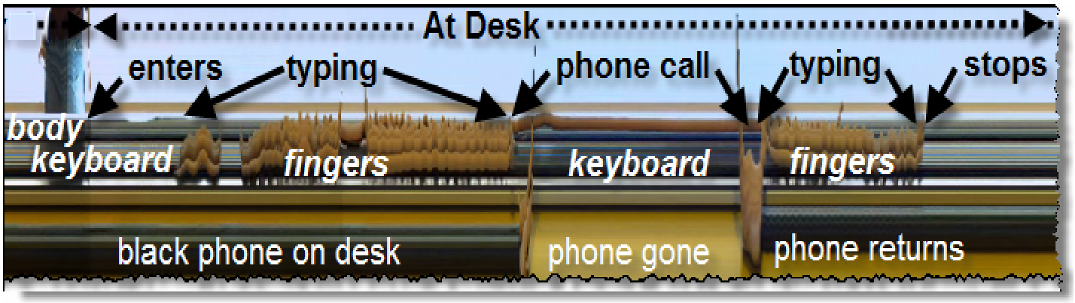


Figure 6b. The minute row captured by the camera angle of Figure 6a.

client keeps in memory only the frames that have a 1:1 correspondence with the slices currently displayed. To do this, the client contains four separate objects representing each of the four video rows (minute, hour, day, and week). Each object independently samples the input stream, and only stores the ones that are needed in an array as an uncompressed frame. Each row object has a different sampling rate – as previously mentioned, the minute object samples much more frequently than (say) the week object. For pragmatic reasons, we chose a constant frame array size of 1020 for each row, and a frame size of 320 * 240 pixels in dimension – a common format for webcam streams. This provides a 'reasonable' balance between frame rate for the minute row (17 fps * 60 seconds = 1020 frames / minute) and memory use (1020 frames * 320 * 240 pixels/frame * 24 color bits/pixel ~= 225 MB / row or 900 MB across all rows). In addition to the currently visible frames, each line also stores a single 1020 * 240 pixel image that represents the currently visible video slice visualization itself. TIMELINE then resizes this image to fit a window, regardless of that window's resolution.

As we go from the hour to the day and the week rows, the interval between sampling successive frames lengthens. If we merely select a frame at random from this interval to display, we may miss significant though brief events occurring within the other frames, something that is increasingly likely as the interval lengthens. TIMELINE uses a change detection algorithm to select the 'best' frame from a series of frames in an interval; the heuristic is that the frame whose pixels differ the most from the previously displayed frame is most likely to contain a significant event. When the sampling interval elapses, the row adds that frame to its frame array, and that row's video slice image is updated by shifting it left one pixel and drawing the appropriate column from the new frame in the rightmost column.

When people move the focus slit, each row rebuilds its video slice image by walking through its frame array, extracting and compositing the appropriate column along the way. When the user scrubs over a visualization line, TIMELINE merely retrieves the frame underlying a selected slice from the array and displays it in the floating window. While we could have compressed or stored these frames on disk, there would simply be too much lag in reading and decompressing the frames to interactively perform these operations.

Each row also keeps an archive of video frames on disk; this archive allows previously seen frames to be displayed when a person wants to retrieve details from the distant past. Frames are stored as MPEG4 v2 compressed video files that are in 1020 frame numbered segments, i.e., that match a unit of frames that can fit in a row. As a row samples new frames and adds them to the array, old ones are removed and sent to a temporary storage to construct the video files. When this temporary storage has been filled by the necessary 1020 frames, the segment is written to disk and removed from memory. By keeping separate archives for each row rather than a single monolithic archive, we are able to speed up the process of retrieving region details in the distant past for display.

For example, consider what would happen in the hour row when a person selects a prior time period in the day row to display in detail, as in Figure 4. If the system used a monolithic archive, it would have to fill the hour line by decompressing, analyzing, and selecting frames from 60 different minute-long 1020 frame segments. By using separate archives, the system has to read and decompress between a minimum of one (if the time matches exactly) to a maximum of two segments (if the time overlaps both) in order to pull in the 1020 frames for the hour row. Even so, the operation is expensive: there is a noticeable delay (typically a few seconds) to rebuild all retrieved row images from disc.

As a side note, we are careful to retain a copy of incoming video even when people are reviewing sequences from the distant past. We do this by creating a new video row object with its own frame array and slice image to hold the archive data as it is read in; the original row objects keep recording incoming frames into memory as before. This is the simplest way to ensure that live video capture continues normally as a person explores the archived data. Additionally, this allows us to instantaneously switch back to the live visualization.

TIMELINE is memory intensive. As mentioned, it uses about 900 MB for the basic fully populated display. When it reads in from the archive, it is adding to what is already stored in memory: up to a maximum of around 2.2 GB if people request archives from the week view (as this reads in all other rows). This amount of memory consumption is feasible on modern high end workstations, and is appropriate to demonstrate the power of such a system. Yet it is unreasonable for low-end machines. Memory requirement could be brought down significantly by reducing the frames per line, the frame resolution, or frame bit depth. The tradeoff is a lower-quality visualization. Still, Moore's Law will make our current parameters reasonable for average machines of the near future.

Related Work

Photography and Interactive Art

The idea of using slit-scans to capture people's activity evolved as a method in photography, film and interactive art installations. There are far too many examples of its use to cover here. Instead, the reader is referred to Golin Levin's (2006) compilation of slit-scan artworks[1] for an excellent overview.

[1] We are indebted to Golin Levin for creating this valuable resource. Many of the examples shown here are extracted from his collection.

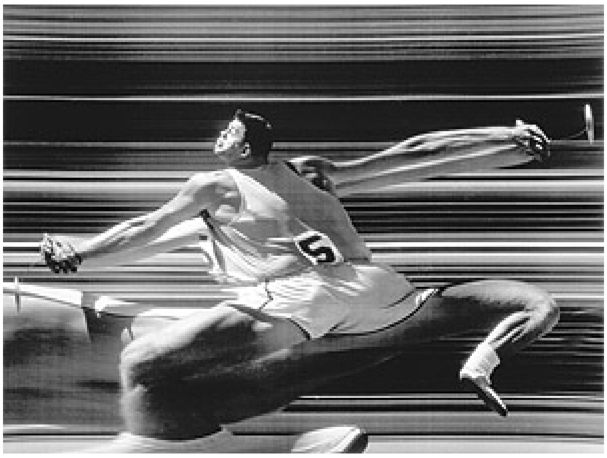


Figure 7. Hammer thrower by George Silk. Reproduced from National Gallery of Australia gallery of Silk's work: www.nga.gov.au/Silk/Gallery.htm

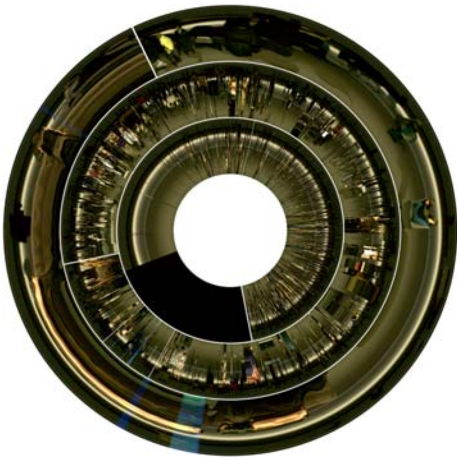


Figure 8. LAST CLOCK. Reproduced from Angesleva and Cooper (2005).

Beginning with photography, slit-scan methods were historically used for creating distortions: a fine slit was moved past the film as the picture was being exposed (Davidhazy, 2007). An example photo done this way is shown in Figure 7: George Silk's Hammer Thrower from the U.S. track team Olympic tryouts, published in Life magazine on July 18, 1960.

Various art projects extend this idea to live digital video. For example, Romy Achituf's Pixel Present (1998) used slit-scan to capture and display live digital video of audience members walking by a large screen. However, the closest visualization to our work is Angesleva and Coopers' (2005) LAST CLOCK (Figure 8). As with other artists, they use video to capture slit-scan images of people moving around an area. However, they stretch the idea of time by fashioning the visualization as a series of concentric rings – a clock – that shows 12 hours of footage. There are three rings: the outer one is the composite image of the last 60 seconds, the middle of the last 60 minutes, while the inner ring of the last 12 hours. New slit scans replace the old, so the points where the images are replaced become the second, minute and hour hand of a clock. Figure 8 illustrates an image from their show at the ACA Media Arts Festival in Tokyo. The white lines in each ring shows that the time is 5:45:55.

Unlike TIMELINE, most approaches to slit-scan art (including LAST CLOCK) create a static image that cannot be explored further. To quote: "Slit-Scan imaging techniques are used to create ***static images*** of time-based phenomena" (Levin 2006, emphasis added). However, few artists allow rudimentary navigation. ARTIFACTS OF THE PRESENCE ERA (Viegas et. al. 2004) used a rock formation metaphor to construct the visualization. Rows of composite slices taken over time are layered atop one another, where older layers were visually flattened and compressed to mimic strata. Visitors could crudely navigate between layers: turning a knob would display a single video frame representing an entire layer.

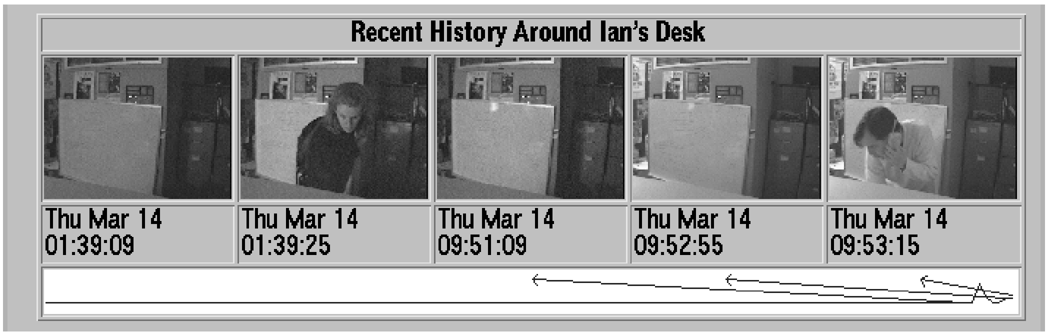


Figure 9. WHEN DID KEITH LEAVE, from Hudson and Smith (1996).

frame), and then navigate the shape by mousing over it. Fels, Lee and Mass (2000) extended this idea by letting people slice through the volume at any angle and position. Again, their purpose was art: “The main purpose is to explore some of the aesthetics of looking at video data from a variety of perspectives’ (p. 370). None of these systems have the level of interactivity provided by TIMELINE.

Video Media Spaces.

A handful of researchers in Computer Supported Cooperative Work have experimented with the notion of somehow capturing people’s activity over time, and using this information to augment a media space. The original motivation for such traces was first raised by Hudson and Smith in 1996, where they saw it as a way “to provide a more general idea of recent patterns of activity without requiring the constant attention of the receiving user” (p. 255).

One approach uses activity graphs: video frames are analyzed for differences, and a graph visually portrays the amount of change in the video over time. For example, an empty office will be seen mostly as a flat line, a person entering or leaving will be seen as a spike, while a seated person will be seen as a wavy line reflecting that person’s small motions. Hudson and Smith (1996) introduced such activity graphs as a way to augment a media space, while Lee, Girgensohn and Schlueter (1997) recommend using such graphs *instead* of video transmission as a way to include people who refused to use video due to privacy concerns.

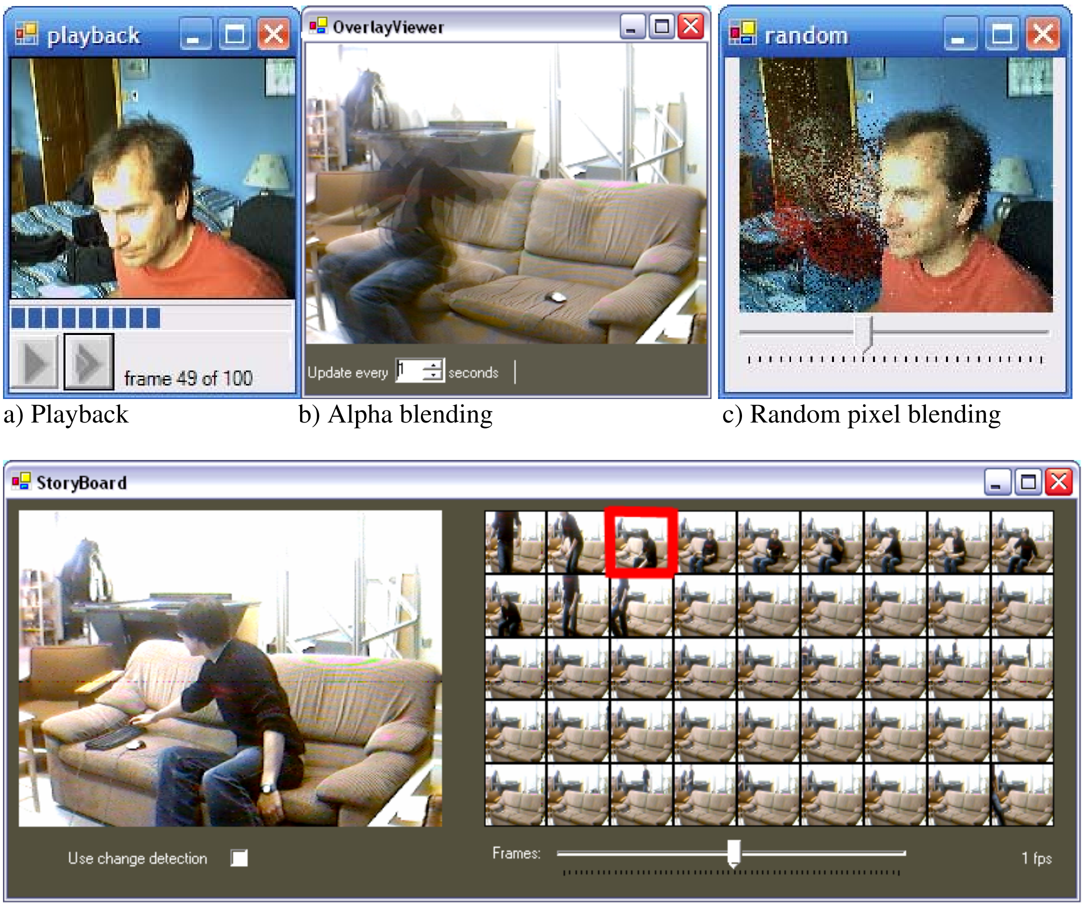
Hudson and Smiths’ (1996) WHEN DID KEITH LEAVE? uses multiple video frames to show activities over time (Figure 9, top) as well as an activity graph (Figure 9, bottom). The system collects a small series of still images over time that characterize the flow of activity in the space. Their algorithm selects and displays a small number of frames (e.g., five) from a video stream, where the chosen frames are those that show significant visual differences in activity. Whenever a new frame is captured, it adds it to the series (while removing the oldest one) when at least 20% of the new image had changed. Thus the series of images need not be linear with time. Gutwin (2002) provides an alternative to this visualization, where he suggested (but did not implement) a short-term video trace where several video snapshots taken in the recent past are alpha-blended onto the current video frame. Hudson and Smith (1996) advocated a somewhat similar technique except that changes were shown as shadows, thus showing some activity (darkness = movement) while masking its details. In all these works, the concept of video traces was a side issue that was not explored in depth: Gutwin (2002) was investigating traces as a way to mitigate network issues in telepointer tracking, while Hudson and Smith 1996 were primarily focused privacy in VMS.

In parallel with video traces, a variety of other researchers try to extract and visually portray patterns collected from computer usage logs and from physical sensors. For example, Begole et. al. (2002) describe how they generate rhythms of personal activities as actograms by analyzing and modeling a person's computer activity, their mail, phone and instant messaging use, and their online calendar appointments. Fogarty, Hudson, et. al. (2004) argue that sensor data can be used to generate models that differentiate between interruptible and non-interruptible situations. While their purpose is to use this data mostly for automating whether a person is interruptible at a particular instance in time, the same data could easily be used to generate temporal patterns of activity.

Our Early Investigations

Our own investigations prior to building TIMELINE also led to several alternate visualizations of activity history. Our first example, rapid playback, was inspired by Dietz and Yerazunis's (1991) idea of letting momentarily distracted cell phone listeners play back short segments of missed conversations: A sensor detects when the phone is moved away from the ear, and records the audio at that point; when the phone is moved back to the ear, the pitch-preserved audio is played back at higher speeds. Similarly, our video-based approach, illustrated in Figure 10a, displays the current image while automatically capturing the last *n* video frames (we used 100 frames captured at 1 frame every two seconds, or 2 1/3 minutes of past activity). The viewer can at any time elect to play back this captured video stream at 20x or 40x normal speed (i.e., as a 10 second or 5 second movie) by pressing a particular speed replay button (Figure 10a, buttons on bottom left – this screen capture is in the middle of rapidly replaying the sequence). This effect works quite well for quickly reviewing the very recent past, and can be extended to include longer time periods and different frame rates. It is limited because no overview is given of past activities, and the person has to actively decide to review the video.

To give some hint of past activity within the visible image, we then realized Gutwin's (2002) unimplemented idea of alpha-blending several frames from the recent past onto a single video frame, as illustrated in Figure 10b (as mentioned, this is a variation of Hudson and Smith's 1996 Shadow method). The further into the past, the more faded a change appeared. As before, a person could set the



a) Playback b) Alpha blending c) Random pixel blending

d) Displaying past frames as storyboard miniatures.

Figure 10. Other earlier approaches tried by the authors on visualization a video trace

frame rate. While this did give a sense of activity, it did not satisfy. It did not scale well beyond a modest number of frames (as the changes in the alpha blended images proved too translucent). If the frame rate was high, so was the glimpse into the past (i.e., activity was usually realized as motion blur). If the frame rate was low, then the composite images were disjoint and hard to interpret. This technique also gives only a short glimpse into the very recent past.

We then tried a variant of alpha blending that we call random pixel blending. We overlayed a percentage of randomly chosen pixels from the just-taken video frame onto the displayed image (Figure 10c). The actual percentage is specified via the slider. If the percentage is small (e.g., 15%), then motions are realized as a pixel scatter blur effect. If a person remains somewhat still, details are slowly filled in. Because backgrounds change rarely, they come in at full view. For example, Figure 10c shows the same image as in Figure 5a. The telecommuter is sitting fairly still behind his computer – thus his image (as well as the room background) is more or less complete. His wife just walked in and gave him a kiss on the cheek; because she is moving, this is seen as scattered pixels as she came through the doorway (top left). A hint of her clothes color is just visible (red shirt and blue jeans) at the bottom left, as is the top of her head as she leans over to kiss him (middle left). Thus privacy is somewhat protected: people are aware of stable events, but details and rapidly done actions are obscured. Still, as with alpha blending this only gives a short glimpse into the very recent past.

Next, Figure 10d illustrates our storyboard display, a variation of Hudson and Smith's "When did Keith leave" approach (1996). An end user sets a sample rate for extracting frames from the video stream, and the last *n* samples are displayed as miniatures in a visual storyboard. For example, Figure 10d has a sample rate of 1 per second; thus the 45 miniatures in the storyboard reveal the last 45 seconds of activity. If it was set at (say) 1 sample per minute, the storyboard would reveal the last 45 minutes of activity. As with TIMELINE, our system displays the frame – chosen from the many generated in that interval – that differs the most from the last storyboard sample. New samples also over-write old ones in the wrap-around sequence (the latest sample is outlined in red, i.e., the 3rd image in the top row in Figure 10d). Compared to "When did Keith leave", this approach conserves time as a true linear stream, and the smaller low resolution images adds some privacy protection. Yet our own impressions of this storyboard technique were not favorable. Because of the many images, it was hard to tell 'at a glance' what was going on. As well, setting a high sample rate (e.g., 1 sample per second) proved distracting and did not provide a long enough history window to justify the screen space. Yet setting a low frame rate (e.g., 1 sample per minute) omits multiple key events that could happen within the sampling period, e.g., 2 visitors arriving and leaving within a few moments of each other.

Reflection

This paper has motivated why visual traces can provide awareness leading to casual interaction, and described TIMELINE as a very efficient method for seeing and reviewing past events. We placed TIMELINE within the context of prior work, provided enough details of its implementation so that others can replicate it, and also make it freely available for others to try.

Yet our motivation is not to provide TIMELINE as a solution. Rather, we want to use it as a case study that pushes the extremes of what is possible, and to provoke debate about video trace technologies within the CSCW community. In this spirit, we reignite the debate started by Hudson and Smith (1996) and Begole et. al. (2002) by offering preliminary reflections on TIMELINE's use. We base our reflections on the reactions of many people (including ourselves) to the Timeline system through numerous live demonstrations and through self-trials. First, we ran TIMELINE as a video mirror within a public interactive installation. The installation was located in a highly visible public part of our research laboratory on a 60" touch-sensitive plasma display (using a webcam attached to this display), and hundreds of visitors saw and tried it as part of several open-house days. Second, we used TIMELINE ourselves, where we ran it for extended periods of time on personal workstations within our laboratory as well as at home.

First, the positive side. The live installation verified that the visualization is easily readable. With only a brief introduction to how it works, visitors were rapidly able to comprehend and read the visualization display, especially because they could see how their actions were immediately represented within it. They were able to spot themselves in the visualization, pick out events such as the arrival of a crowd for a demonstration session, and see the rhythms of activity between night and day. People found it easy to reason about what the display was showing. They could fine odd events of interest, such as a brief period of light in the middle of the night. Scrubbing and moving the slice focus bar let people determine how patterns in the visualization were created, enhancing their understanding as well as their ability to read other parts of it. While static images produced using slit-scanning techniques can appear to be distorted and strange, augmenting them with this level of interactivity brings them into comprehension.

People thought they could be playful when using TIMELINE as a history mirror. This replicates what has been seen before in slit-scan art installations. People made patterns within the visualization by waving arms or colourful items, or even slowly turning around in front of the camera to produce a flat scan of their head all the way around. Visitors would often pose in front of the installation to leave their mark in the visualization (also noted by Viegas et. al. 2004). People commented on the aesthetics of the visualizations in showing clearly the cyclic rhythms of activity and how changes occur over time within a space. People also found scrubbing compelling as a form of voyeurism; they really wanted to see what other visitors had done.

While the public demonstrations were set up as an interactive installation, people also suggested practical uses for it. Those who did not know about the awareness motivation typically said that TIMELINE had great potential as a surveillance system used for security purposes. Their ideas ranged from its use by trained security personnel, to home monitoring (e.g., nanny cams), to property protection when away, and to equivalents of baby monitors. Other ideas included using TIMELINE as a research analysis tool for detecting patterns and counting key events in captured video streams.

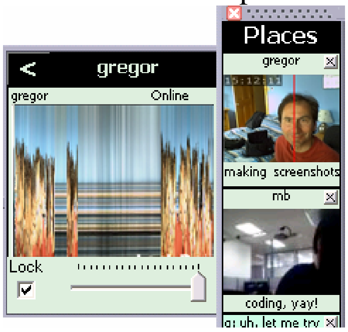
Now we turn to the dark side. Privacy is an extremely serious issue as TIMELINE does its job too well. Hudson and Smith (1996) believe systems such as these demonstrate how privacy issues might occur when providing awareness information through a video trace: transient activities of a person are no longer lost if the system records and displays significant events. TIMELINE's power could make it very invasive. Potentially embarrassing, private or inappropriate behavior is not only captured, but easily found and replayed. Thus a trade-off arises: while collaborators can benefit by using the system to see each other's recent activities and events, it also serves as a (perhaps unintended) surveillance system (see Boyle and Greenberg, 2005b for a comprehensive discussion of privacy issues in VMS).

Our own uses of the system reflect these concerns. Even as creators we were reluctant to use it in broadcast mode; most of our experiences have been using it as a local mirror. Still, we were willing to try it within our research laboratory, an open space that included other workers. Yet this immediately incited concerns from other inhabitants of the laboratory, for they were very uncomfortable with the idea of being permanently captured on video for others to see (even though all actions were in a public space). There were meetings and discussions about this, and eventually people were willing to have it run in very limited situations because they valued its use for research. Even so, we found that we inadvertently captured other people in embarrassing situations, e.g., a cleaner who went to sleep on a lab couch during a night work shift.

Similarly, one author telecommutes from his home, and regularly uses a traditional VMS. He was willing to use TIMELINES in work situations with his distant colleagues, but was concerned that other family members using the home office (which also doubled as a spare bedroom) would be unintentionally captured. This would not only be embarrassing for a family member if caught in a compromising situation, but also for the distant viewer who could unintentionally see that situation in TIMELINE.

Another downfall is that TIMELINES also allows people to very easily compare their activity with others (if multiple instances are running), and in turn they may become overly self-critical of their own work habits. Tacit information becomes explicit. The system allows observations such as "did I work as many hours as my colleagues?" This could lead to undue stress on the workers' part, which would be magnified if the worker knew their video stream was being broadcast.

Still, there are ways to use TIMELINE that may reduce privacy invasion. For example, adjusting the camera angle can make a large difference in what is captured, e.g., Figure 6a+b shows how pointing the camera at a desk highlights when a person is at their computer and/or on the phone, but does not transmit head shots or surrounding areas. Similarly, positioning the camera outside an office doorway (as done by Buxton 1997) means that the remote viewer can only see what people walking by a hallway can see, and that closing the door protects oneself in both the physical and virtual space. Other methods developed for safeguarding privacy in VMS can also be used, e.g., context-sensing for controlling when video is taken (Neustaedter and Greenberg 2003) or image blurring in low-risk situations (Boyle and Greenberg 2005b). There is also the issues of distraction with TIMELINES full-screen view. This can be solved by embedding it into more traditional media spaces. For example, the inset shows it redesigned fit within Community Bar



making screenshots

coding, yay!

g: uh, let me try

(McEwan and Greenberg 2005). People see a traditional media space view in a side bar, but can raise a video trace by mousing over it for an overview of hourly activities. Clicking this trace then raises the full TIMELINES system.

TIMELINES is best viewed as an investigation into the extreme, where it asks the question "what would it be like if we could easily see and explore a video trace of a distant collaborator'? Currently, it is difficult to predict where it might be best used and the cultural practices that would evolve around it. Perhaps TIMELINES could be suited to areas that are largely accepted as public rather than private offices and workspaces. Perhaps it would work well between tight teams or social intimates with a strong desire to stay connected. Perhaps it can be used as a base reference to other approaches, e.g., to see if low fidelity actogram graphs (Begole et. al. 2002) provide comparatively sufficient awareness information while still safeguarding privacy. Or perhaps CSCW is the wrong venue; it may better serve as a security system, as art, or as video analysis. This is clearly an area for future – and perhaps controversial – research. Let the debate continue.

Acknowledgments

This research was partially funded by NSERC through its scholarship, Discovery Grant and NECTAR Networks grant program, and by Alberta Ingenuity and iCORE. We also thank our industrial sponsors: Microsoft Research and Smart Technologies, Inc.

References

Achituf, R. (1998): *Pixel present: A gestural scanner.'* Accessed March, 2007. http://www.gavaligai.com/main/sub/installation/PixelPresent/PixelPresent.html

Angesleva, J. and Cooper, R. (2005): Last Clock. IEEE Comput. Graph. Appl. 25, 1, 20-23.

Nunes, M., Greenberg, S., Carpendale, S. and Gutwin, C. (2006): 'Timeline: Video Traces for Awareness'. *Video Proc. ACM CSCW*'06, ACM Press.

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 '02*. 334-343.

Bly, S. Harrison, S. and Irwin S. (1993): 'Media Spaces: Bringing people together in a video, audio, and computing environment', in *Comm. ACM*, 3, 1, 28-47.

Boyle, M. and Greenberg, S. (2005a) Rapidly Prototyping Multimedia Groupware. Proc Distributed Multimedia Systems (DMS'05), Knowledge Systems Institute, IL, USA.

Boyle, M. and Greenberg, S. (2005b): 'The language of privacy: Learning from video media space analysis and design'. ACM Trans. Computer-Human Interaction (TOCHI). 12 (2), June, 328-370.

Buxton, W. (1997): 'Living in augmented reality: Ubiquitous media and reactive environments'. In K. Finn, A. Sellen & S. Wilber (Eds.). Video Mediated Communication. Hillsdale, N.J.: Erlbaum, 363-384.

Davidhazy, A. (accessed March, 2007) Slit-scan photography. School of Photographic Arts and Sciences, Rochester Institute of Technology. http://www.rit.edu/~andpph/text-slit-scan.html.

Dietz, P. H. and Yerazunis, W. (2001): 'Real-time audio buffering for telephone applications'. *Proc ACM UIST*. 193-194.

Elliot, E. (1993): *Watch * Grab * Arrange * See: Thinking with motion images viea streams and collages. MS Thesis, Master of Science in Visual Studies, January, MIT.*

Fels S., Lee, E. and Mase, K. (2000): 'Techniques for Interactive Video Cubism'. Proc ACM Multimedia. 368-370.

Fogarty, J., Hudson, S. E., Atkeson, C. G., Avrahami, D., Forlizzi, J., Kiesler, S., Lee, J. C., and Yang, J. (2005): 'Predicting human interruptibility with sensors'. *ACM TOCHI* 12, 1 (Mar), 119-146.

Gutwin, C. (2002): Traces: Visualizing the immediate past to support group interaction. *Proc Graphics Interface.* 43-50.

Hudson, S. E. and Smith, I. (1996): ‘Techniques for addressing fundamental privacy and disruption tradeoffs in awareness support systems’. *Proc ACM CSCW*. 248-257.

Johnson, B. and Greenberg, S. (1999): ‘Judging People’s Availability for Interaction from Video Snapshots’. *Proc. HICSS Distributed Group Support Systems Minitrack*, January, IEEE Press.

Kraut, R., Egidio, C., Galegher, J. (1990): ‘Patterns of contact and communication in scientific research collaboration’. *Intellectual Teamwork: Social and Technological Foundations of Cooperative Work.* LEA

Lee, A., Girgensohn, A., and Schlueter, K. (1997): ‘NYNEX portholes: initial user reactions and redesign implications’. *Proc ACM GROUP*. ACM Press, 385-394.

Levin, G. (2006) ‘An informal catalogue of slit-scan video artworks’. Last edited December 29, 2006. http://www.flong.com/writings/lists/list_slit_scan.html.

McEwan, G. and Greenberg, S. (2005): ‘Supporting social worlds with the Community Bar’, *Proc. ACM Group* 21-30.

Neustaedter, C. and Greenberg, S. (2003): ‘The design of a context-aware home media space. *Proc UBICOMP’03*, 297-314, LNCS Vol 2864, Springer-Verlag.

Viégas, F. B., Perry, E., Howe, E., and Donath, J. (2004): ‘Artifacts of the Presence Era: Using Information Visualization to Create an Evocative Souvenir’. *Proc IEEE Infovis'04*. 105-111.

Whittaker, S., Frolich, D., and Daly-Jones, O. (1994): ‘Informal workplace communication: What is it like and how might we support it?’ *Proc ACM CSCW*, 131-138