Support for Effective Use of Multiple Video Streams in Security

Andreas Girgensohn[1], Frank Shipman[2], Anthony Dunnigan[1], Thea Turner[1], Lynn Wilcox[1]

[1]FX Palo Alto Laboratory, Inc.
3400 Hillview Avenue
Palo Alto, CA 94304, USA

[2]Department of Computer Science &
Center for the Study of Digital Libraries
Texas A&M University
College Station, TX 77843-3112

{andreasg, tonyd, turner, wilcox}@fxpal.com, shipman@cs.tamu.edu

ABSTRACT
Video surveillance systems have become common across a wide number of environments. While these installations have included more video streams, they have been also placed in contexts with limited personnel for monitoring the video feeds. In such settings, limited human attention in combination with the quantity of video makes it difficult for security personnel to identify activities of interest. Furthermore, interrelationships among activities in different video streams are difficult to ascertain. We have developed applications to support security personnel both in analyzing recorded video and in monitoring live video streams. We use a variety of analysis techniques to determine unusual events and to highlight them in video images. Our main focus has been on visualization techniques and user interfaces that direct the attention of security personnel to the most important activities within recorded video or among several live video streams.

Categories and Subject Descriptors
H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems – *video*. I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis – *motion, tracking*.

General Terms
Algorithms, Design, Human Factors.

Keywords
Video surveillance, multiple video streams, security cameras, video summary.

1. INTRODUCTION
Video surveillance systems are common in commercial, industrial, and residential environments. However, limited human attention and the number of video streams constrain the cost efficiency and effectiveness of such systems. Many of the larger installations have well over 1,000 video cameras. Most video surveillance systems today are passive collectors of video streams, presenting a small subset on monitors for review by security personnel. Monitoring personnel have difficulty effectively focusing on video for long stretches of time. While many of the older installations use analog video, the use of digital video provides opportunities for proactive video systems that alert the users of important events. Technology can be used to help pinpoint important activities so that personnel can focus on the most relevant segments of video.

Much of the current research in video surveillance focuses on algorithms to analyze video and other media from multiple sources to automatically detect significant events [8]. However, automatic algorithms do not always correctly identify events so that keeping the human in the loop is crucial. We also have developed algorithms for the analysis of live and recorded video from multiple cameras in multiple locations but the main focus of our work is to provide user interfaces that direct the attention of security personnel to the most important activities. Our approach differs from traditional security monitoring systems in that it draws the attention of the security personnel to the video that is likely to show interesting events. Commercial system have capabilities such as "on event" recording and support the input of "rules" for raising alarms that display video streams [10, 16]. Our goal is to support security personnel both in analyzing recorded video as well as in watching live security video for important incidents. The approach taken combines several techniques that together provide an interface for quickly ascertaining the activity across a number of cameras during a period of time.

The details of our techniques are described in the remainder of the paper. Before doing so, we examine prior techniques and related work. Then, we describe several approaches for determining security incidents quickly and discuss additional approaches for detecting more complicated events. We present different ways to visualize activities in keyframes. Next, we describe two different approaches for summarizing events in the form of timelines or storyboards. Finally, we show a multi-stream video player that gives users access to live and recorded video and that automatically draws the users' attention to important events.

2. RELATED WORK
There are many commercial systems for video surveillance. Given ideal video conditions and little or no visual confusion, commercial video surveillance systems with video analytics available today can detect if persons or vehicles present and possibly iden-

VSSN'06, October 27, 2006, Santa Barbara, California, USA

tify them, find objects are left behind, detect activity occurring in restricted areas or certain limited behaviors. Security personnel can then be alerted with the relevant video brought up on a monitor and sometimes, with boxes or circles drawn around the object that triggered the alarm. A panel at ISC West 2006 [15] identified almost 40 commercial systems that involve some form of video analytics.

For detecting events and for visualizing activities, it is useful to separate foreground and background pixels in video streams. To accomplish this, several researchers have applied Gaussian mixture models [6, 27].

Timelines lend themselves to be used for navigating videos and for visualizing time-based events. They have been explored by a variety of researchers. Plaisant et al. [19] use timelines to visualize events in people's lives (e.g., criminal or medical records). Kumar et al. [14] visualize data from digital libraries such as information about music composers in timelines.

For video and image summaries, different keyframe approaches have been used. Taniguchi et al. [21] described a 2-D packing of “panoramas” for video browsing. Daniel and Chen [9] summarize video with a volume visualization technique. Yeo and Yeung [25, 26] create video summaries based on a dominance score. They place images of different sizes that are integer multiples of a base size in a grid. PhotoMesa [4] presents hierarchies of images of similar sizes. It uses Quantum Strip Treemaps that fill a 2-D area by partitioning a rectangle into sub-rectangles that in turn can be partitioned further.

Huston et al. [11] present a system that ties views from several cameras together with a map. Their system only displays a single video player and does not vary the size of video players and keyframes to indicate importance. Iannizzotto et al. [12] also present a system combining a map with views of multiple cameras but their paper mostly focuses on gesture input. Rather than using a 2D map, Sebe et al. [20] map objects detected in video onto a view of a 3D environment by distorting the objects appropriately. We decided against the use of 3D views because they are frequently confusing to the user.

3. IDENTIFYING ACTIVITY IN VIDEO

While very elaborate approaches for analyzing video have been described in the literature, our requirement is that the analysis can be performed in real-time for tens or even hundreds of cameras. It is possible to divide the analysis among several computers but it is not cost-effective to require a separate computer for every single security camera. This requirement limits the sophistication of our analysis.

We model the background of the camera view and determines foreground pixels in every video frame. Other approaches are given in [6, 7]. For the background separation, to avoid a second pass and to be able to deal with live video, we only consider video frames earlier than the current frame. We determine the median value for each pixel by computing a histogram of all values for that pixel position. Thus, we do not have to store all previous pixel values in memory. We favor recency using an exponential decay factor for the histogram where older values have a lower weight than newer ones. This approach deals very well with video noise but has problems with sudden or gradual shifts in lighting conditions.

Once foreground pixels have been identified, they are used to characterize the degree of activity in the video. Sequential frames with activity rates above a preset threshold are grouped first by major changes, such as lights being turned on or off. Within these major video groupings, we determine activity segments independently using thresholds for the minimum fraction of pixels changed to be considered valid activity, for the minimum pause in activity to start a new segment, and for the minimum length of an activity segment to ignore video noise. Importance is assessed based on the amount of activity in the video and on motion close to pre-specified points of interest in the space being videotaped.

While our events currently are at a fairly low level (e.g., activities near a hot spot), we are currently working on a more sophisticated event recognition such as the one described by Atrey et al. [2] or Wu et al. [24] and on a sensor network similar to the one described by Ardizzone et al. [1].

4. VISUALIZING ACTIVITY IN KEYFRAMES

We use keyframes to visualize segments of activity when automatically summarizing events in recorded video streams. There are many situations where a user needs to get a sense of activity in a video segment. Security personnel do so when determining if a video segment considered interesting by the system is actually of interest. Due to limitations in bandwidth, screen real estate, and the ability to assess multiple video segments simultaneously, interfaces other than video players are needed for presenting activities in a video segment from one or more cameras. We present three approaches meant to provide users with an understanding of video by grouping keyframes or by overlaying keyframes with visualizations of activity. We also discuss how user interaction with these visualizations can select different portions of the video.

4.1 Keyframe Compositions

Our initial approach to presenting simultaneous action in multiple video streams was to group keyframes into piles (see Figure 1). The user can flip through the pile by moving the mouse over the borders of obscured keyframes. Piles can either group keyframes from different cameras at the same time or from the same camera at different times.

An alternative to keyframe piles is a composite image generated by combining areas of interest in keyframes from multiple cameras. The keyframe composition shown in Figure 2 uses the size of the regions taken from the source keyframes to indicate the relative importance of activity in those video streams.

4.2 Overlays of Activity

A second approach to presenting video activity in a still image is to overlay a visualization of activity on a keyframe. Figure 1 illustrates such a visualization wherein a colored halo is drawn around the shapes identified as foreground pixels. To deal with video noise and to make the shapes smoother, background pixels that are surrounded by foreground pixels are included in the visualized object. We also ignore stray foreground pixels. Wang et al. [22] describe a different approach to highlighting regions of importance where they warp a video image to zoom into the important region.

Figure 1. Halo to emphasize the changed pixels.

An alternative form of keyframe overlay indicating activity is to visualize the results of object tracking. The trajectories of objects, or representative parts of objects, (e.g., the highest point, or corners in detected edges), are shown over time as a series of points taken at regular time intervals (see Figure 3). Such markings indicate object motion without being visually cluttered. This approach is useful when multiple objects pass through common regions as different colors can represent different objects or different points in time.

4.3 Time Lapses

Rather than using a more abstract visualization of activity, the third approach uses time-lapse presentations from the actual video. One could naively expect that moving objects can be shown by periodically sampling frames in the video and by creating a single image from averaging the pixels of the video frames. This is the approach

Figure 2. Quad representation of keyframes from four cameras with keyframes cropped to the center of activity and sized proportional to their importance.

Figure 3. Trajectory of tracked object shown as dots marking positions at regular intervals.

used in strobe photography where a strobe light periodically makes an otherwise dark object visible so that it is captured by a camera with a very long exposure. However, this approach only works because the background is dark. A bright background dominates in such a combined frame, and foreground objects are only faintly visible.

We achieve a similar effect by alpha-blending pixels from different video frames. We separate the background (or non-moving objects) from the foreground (or moving objects) and then overlay the foreground on a selected keyframe (see Figure 4). The sample rate of the overlays determines how distinct the foreground objects are. Too frequent samples result in successive overlays of a moving object to blur together, creating a wash across the keyframe. When samples are too far apart, the continuity of action become more difficult to discern. A technique that works best is to select sample rates where moving objects from different frames do not overlap. Our approach produces similar results to the one described by Petrushin [17] where foreground pixels from different times are painted on top of the separated background. However, our alpha-

Figure 4. Alpha-blended foreground with 0.5 samples per second and more emphasis for every fourth sample.

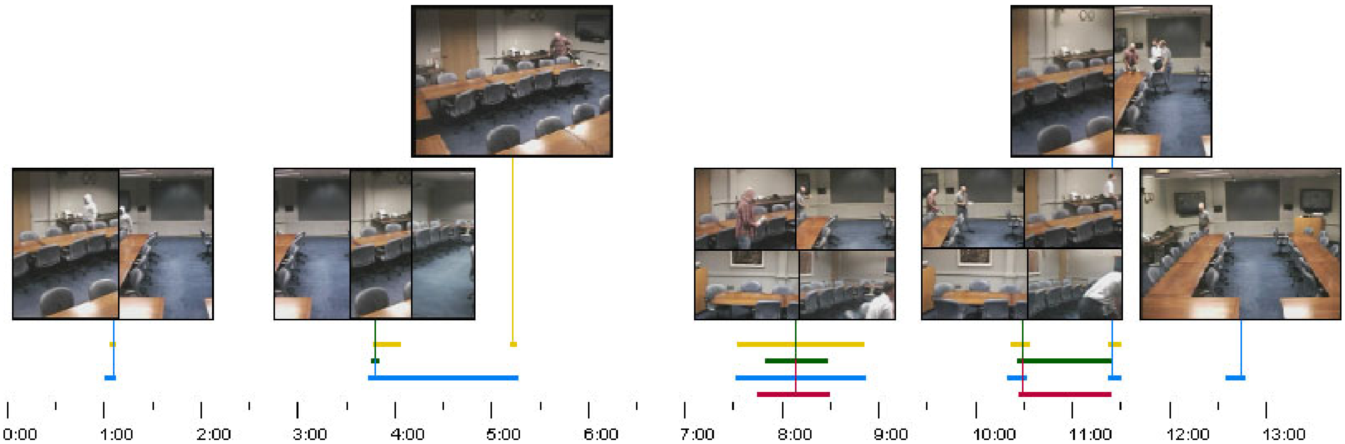

Figure 5. Timeline with events from multiple cameras and event keyframes grouped into quads.

blending approach provides more control over the final result and is less susceptible to video noise.

This time-lapse image can be enhanced by emphasizing foreground pixels periodically (e.g., every fourth sample). This is accomplished by increasing the opacity of those foreground pixels to provide more detail without causing too much overlap in the motion (see Figure 4).

4.4 User Interaction

Keyframe composition, activity overlay, and time lapses are created automatically. However, these activity visualizations can be improved based on user interaction. Users may click with the mouse on either a single video frame or a generated keyframe. If an object is identified near the mouse position, that object is marked as important and tracked across frames. The visualization options described above are then applied to just that object.

Clicking on a keyframe can also be mapped back to a time, either by identifying objects or just by comparing the mouse position to the centroids of the foreground pixels of the sampled video frames. The centroid closest to the mouse position determines the corresponding time. Users may also specify a period of time by dragging the mouse over a region, which is taken as an interval selection defined by the minimum and maximum times associated with centroids in that region. Once the time (or interval) is determined, either the video can be played at that time or the time can be highlighted in the generated keyframe. The latter can be accomplished easily by alpha-blending the video frame at that time with the generated keyframe. When an interval is specified, a new keyframe can be generated that just visualizes the specified interval.

5. EVENT SUMMARIES

We provide two variants of event summaries. The first presents a time-based visual representation of events in the form of a timeline with attached keyframes. The second indicates the relative importance of keyframes by displaying them in different sizes.

5.1 Event Timeline

In addition to the need to understand a period of activity in a still image, security personnel need to understand the temporal relations between periods of activity. Timeline views have been used to provide such an understanding [14, 19]. Within the context of security video, timelines can present a time-based visual representation of events within the field of view of different cameras. As described above, events are identified by determining periods of activities that are considered of interest based on the amount of activity in the video and on distance to points of interest in the space being videotaped. If multiple cameras have the same point of interest in view, the importance measure can be improved by considering distances from all cameras to the point of interest. We designed a timeline display that presents iconic representations of events and indicates the event duration as colored bars (see Figure 6).

We also use a different timeline design shown in Figure 5. Horizontal bars of different colors indicate events for these cameras. Keyframes from overlapping events are grouped together as quads. To choose among possible groupings, the total importance of keyframes selected at the same time is maximized. Grouped keyframes are presented as collages or piles.

During video playback, an indicator on the timeline represents the playback position. Clicking on the timeline or on a keyframe starts the playback of all the displayed video streams at the corresponding position.

5.2 Event Storyboards

Security video applications can also exploit approaches used to summarize video. A common approach is to use a sequence of keyframes that become a filmstrip for the activity. One problem with this approach is that the relative importance of the various frames of activity is not indicated in the presentation. Instead, we use an approach inspired by Manga video summaries [5] that present key-

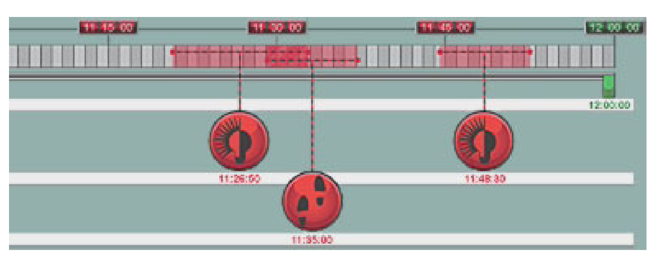
Figure 6. Timeline with attached icons representing different types of events.

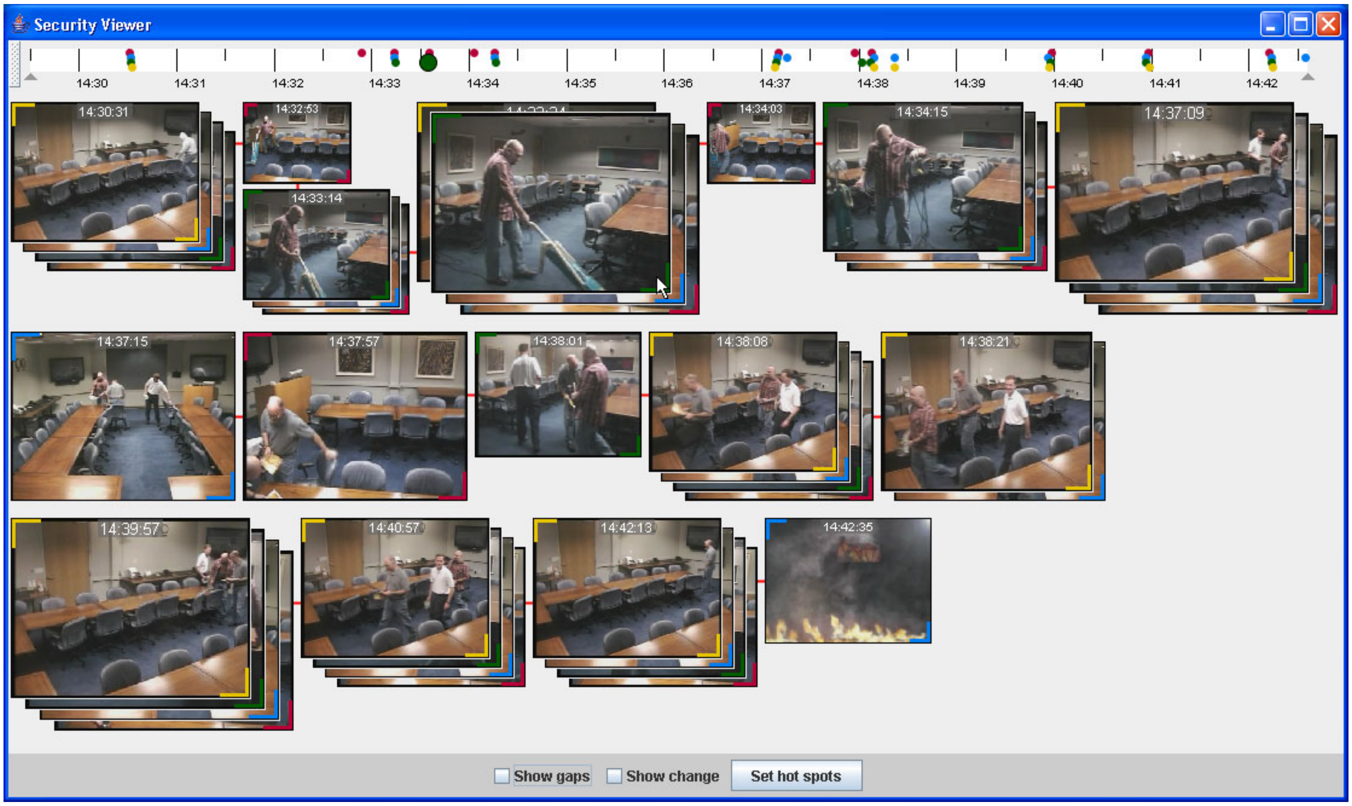

Figure 7. Storyboard from four security cameras with keyframes grouped into piles.

frames in different sizes in the form of a comic strip to indicate the importance of the corresponding video segments.

Manga video summaries were designed to provide interactive storyboards for edited video. As such, their generation relied on the video characteristics of edited video, such as breaking a video up into segments based on where a camera operator or video editor chose to switch among several cameras or video shots. Security video has no breaks and thus techniques to generate summaries of edited video do not work for continuous feed video. Thus, we used a different approach to segment the video and to determine the importance of those segments that work with unedited video. This approach makes use of our techniques for locating periods of interesting activity within a video stream. A second issue with the use of Manga summaries is that they are designed to summarize a single video feed. We need to summarize activity from a set of cameras in the context of security video. To accomplish that, we use the techniques for keyframes compositions described earlier. Finally, the layout algorithm needs to preserve the connection between keyframe size and importance, rather than to resize some keyframes to produce an aesthetically more pleasing summary.

5.3 Creating a Storyboard

To create the storyboard, the video from each camera is segmented into activity segments. Depending on the allocated space for the storyboard, longer activity segments are segmented into several subsegments based on the cumulative amount of activity. In our application, the number of subsegments is proportional to the square root of the cumulative amount of activity to avoid a dominance of busy segments.

The size of a keyframe for a video segment is proportional to the importance of the segment. The Manga algorithm for video summaries changed the importance-specified size in order to generate a storyboard that had the same aesthetics as a manga or comic book (e.g. no blank space). However, in security video applications, it is necessary to accurately depict the importance so that the layout algorithm may not resize keyframes to produce a more compact display. Instead, we use a packing algorithm that inserts whitespace if the keyframes cannot be packed without gaps.

To create a storyboard for multiple synchronized video streams, the algorithm groups together simultaneous segments of activity across several video streams. With overlapping activity segments, groupings are determined that maximize the number of combined video streams that can be visualized by keyframes at approximately the same time. For each group, the keyframes are placed in a pile ordered by descending importance such that the most important keyframe is completely visible and other keyframes can be accessed by mouse movement. Color-coding indicates the source (e.g., camera) of each keyframe. Figure 7 provides the storyboard for four cameras (marked in the corners in yellow, blue, green, and red).

When moving the mouse to reveal keyframes in piles, additional information such as the source camera location can be displayed as a tool tip attached to the keyframe. This mouse movement can also activate a magnifier display that shows the keyframe under the mouse in more detail. Mouse clicks on keyframes start the playback of the corresponding video segment.

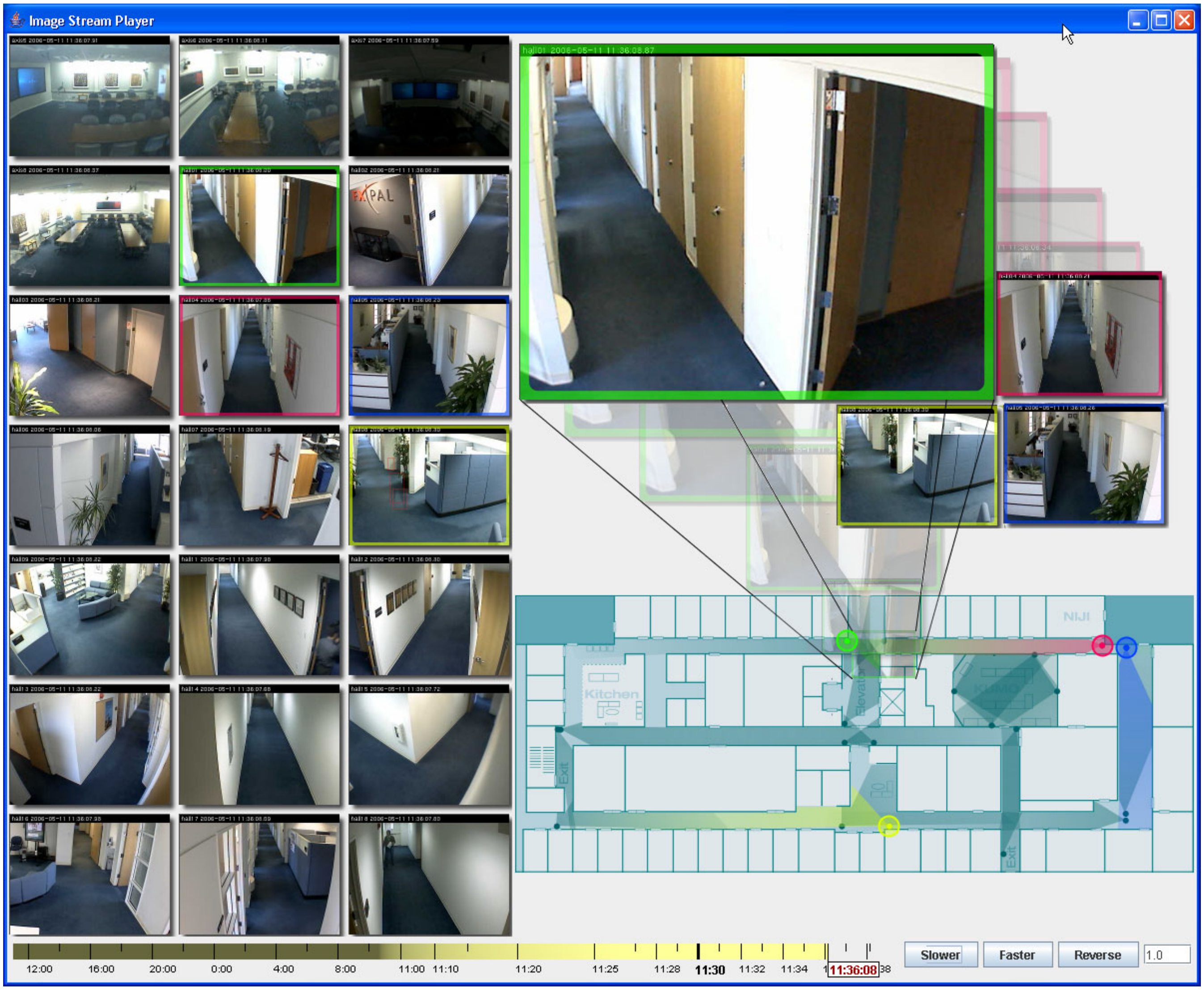

Figure 8. Multi-stream video player. The left side shows a traditional security camera display. In the top-right, interesting video streams are displayed. The map in the bottom-right shows the positions of the currently selected cameras in color. The timeline provides access to both live and recorded video and shows more detail around the selected time.

Storyboards cannot provide much detail when used with many cameras or over long periods of time. For a more detailed view, security personnel can select specific video streams for inclusion in the storyboard and the time frame of the storyboard. We developed map and timeline interface components for these purposes. The map interface is described in the next section as part of the multi-stream video player.

6. MULTI-STREAM VIDEO PLAYER

Based on our above work, we developed a multi-stream video player that combines video displays at different resolutions and frame rates, a map indicating camera positions, and a timeline for navigating through recorded video or switching to real-time video (see Figure 8). The player gives users different options for accessing video by time, by location, or from a low resolution version. The player also provides automatic walkthrough and draws the users' attention to important events.

6.1 Bank of Videos

The left side of the player interface shows a traditional security camera interface. It presents a large number of cameras in low resolution and at a low frame rate. All video displays are synchronized to the same playback position such that skipping to a different position in the timeline will control all video displays. We are currently working on groups of video displays that can show different times, for example leading or trailing the main playback position by a fixed interval. Users can manually select those cameras to see a larger video display at a higher frame rate in the main video playback area.

6.2 Main Player Area

The main player area displays one or more video streams at high frame rates. The size of a video stream display indicates its relative importance. The top-left position is reserved for the most important video streams and video streams are animated out of the way

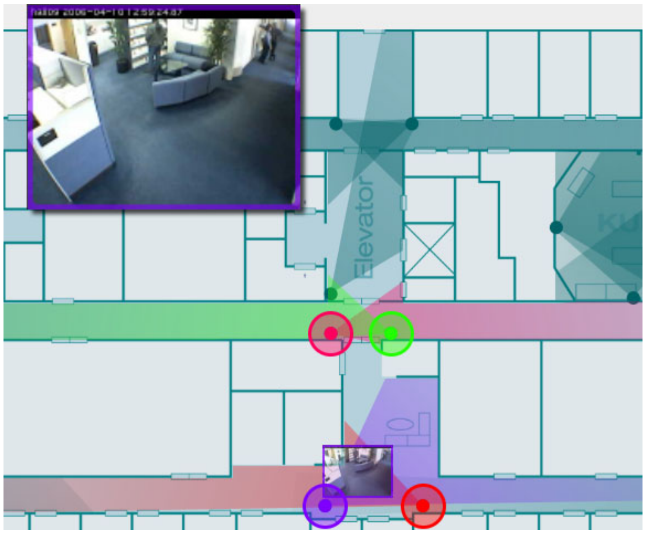

Figure 9. Map showing camera locations and view coverage of selected cameras. The camera image is animated into the player area.

as a new video stream becomes important. The system analyzes the video streams and automatically selects cameras of interest if anything unusual happens. Wang et al. [23] propose to use experiential sampling to switch the view of surveillance cameras but they do not describe a user interface that would use that feature. In the absence of user interaction or unusual events, the system enters a mode similar to a screen saver where cameras are periodically selected in a spatial order resembling the rounds of a security guard.

6.3 Timeline

The timeline at the bottom of Figure 8 provides access to the recorded video and lets the user switch back to live video. It uses a detailed linear scale for the video around the current playback position and a much less detailed linear scale for the video far away from the playback position. A non-linear scale provides the transition between those parts of the timeline. The density of the timeline is color-coded where denser areas are darker. Controls let the user increase and decrease the playback speed and reverse the playback direction.

6.4 Map

Security video installations often monitor hundreds of video streams. Security personnel need to be able to select which video streams to include in the storyboard, the event timeline, or the multi-stream video player. A map interface component has been designed and developed for this purpose.

The map shows the spatial position of the cameras and their fields of view. It is used for orienting to and selecting video streams to include in the storyboard or the video player. Cameras are color coded for identification (see Figure 5). When a user selects a set of cameras with the mouse, the storyboard and timeline are updated, and the camera video streams are displayed in the video player area. Clicking on a map position away from a camera selects all the cameras that can see that part of the map. The map and timeline interact to provide the user with the capability of specifying the locations and times of video segments of interest for more detailed analysis.

When video streams are selected for display in the video player area, the video stream display is animated from the map position to the video player area (see Figure 9). A small keyframe is left behind to anchor the animation to the map. That keyframe slowly fades out.

6.5 Infrastructure

We use Axis IP cameras [3] that provide access to video as Motion JPEG via HTTP or as MPEG-4 via RTP. We decided to focus on Motion JPEG because it has better image quality, is simpler to process, and better supports seeking to different times. Also, Motion JPEG does not require a dedicated codec on the client side so that we were able to build a Java client that animates video player windows. We record two different resolutions (640x480 and 320x240) from each camera at 15 frames per second. The lower resolution saves CPU time and network bandwidth for clients. The cameras could provide 30 frames per second in a single resolution but we decided against the higher storage and network bandwidth requirements. We currently record 22 cameras and store the last 24 hours. Each camera generates about 1.7 GB worth of images per hour (70% for the larger resolution) so that we need about 900 GB disk space and 85 Mbps network bandwidth. We currently work on a approach that can record video at high frame rates and quality and later reduce the quality during less interesting periods. This is similar to approaches described by Korshunov and Ooi [13] and Pillai et al. [18].

7. CONCLUSIONS

Because automatic algorithms do not always correctly identify events, it is crucial to keep the human in the loop. The main focus of our work is to provide user interfaces that direct the attention of security personnel to the most important activities. Our goal is to support security personnel both in analyzing recorded video as well as in watching live security video for important incidents. In this paper, we presented several approaches to support security personnel in dealing with video from multiple security cameras. We provide activity-highlighting video summaries in the form of enhanced keyframes, timelines and storyboards to give users quick access to interesting events in recorded video. For live video, we automatically draw the users' attention to video streams with unusual events by enlarging them and animating them into the center of view. We support users in seamlessly switching between live and recorded video and in synchronizing the playback of many video streams. For both live and recorded video, we connect the video streams to a map of camera locations for better orientation.

REFERENCES

[1] E. Ardizzone, M. La Cascia, G. Lo Re, and M. Ortolani. An integrated architecture for surveillance and monitoring in an archaeological site. Proc. ACM Workshop on Video Surveillance and Sensor Networks, pp. 79-86, 2005.

[2] P.K. Atrey, M.S. Kankanhalli, and R. Jain. Timeline-based information assimilation in multimedia surveillance and monitoring systems. Proc. ACM Workshop on Video Surveillance and Sensor Networks, pp. 103-112, 2005.

[3] Axis Communications. http://www.axis.com/

[4] B.B. Bederson. PhotoMesa: A Zoomable Image Browser Using Quantum Treemaps and Bubblemaps. UIST 2001, ACM Symposium on User Interface Software and Technology, pp. 71-80, 2001.

[5] J. Boreczky, A. Girgensohn, G. Golovchinsky, and S. Uchihashi. An Interactive Comic Book Presentation for Exploring Video. CHI 2000 Conference Proceedings, ACM Press, pp. 185-192, 2000.

[6] S.-C. Cheung and C. Kamath. Robust techniques for background subtraction in urban traffic video. Video Communications and Image Processing, SPIE Electronic Imaging, 2004.

[7] A. Colombari, M. Cristani, V. Murino, and A. Fusiello. Exemplar-based Background Model Initialization. Proc. ACM Workshop on Video Surveillance and Sensor Networks, pp. 29-36, 2005.

[8] R. Cucchiara. Multimedia surveillance systems. Proc. ACM Workshop on Video Surveillance and Sensor Networks, pp. 3-10, 2005.

[9] G. Daniel and M. Chen. Video Visualization. Proc. IEEE Visualization 2003, pp. 409-416, 2003.

[10] Energy Control, Inc. http://energyctrl.com/security.htm

[11] L. Huston, R. Sukthankar, J. Campbell, and P. Pillai. Forensic video reconstruction. Proc. ACM Workshop on Video Surveillance and Sensor Networks, pp. 20-28, 2004.

[12] G. Iannizzotto, C. Costanzo, F. La Rosa, and P. Lanzafame. A multimodal perceptual user interface for video-surveillance environments. Proc. Conference on Multimodal Interfaces, pp. 45-52, 2005.

[13] P. Korshunov and W.T. Ooi. Critical video quality for distributed automated video surveillance. Proc. ACM Multimedia, pp. 151-160, 2005.

[14] V. Kumar, R. Furuta, and R.B. Allen. Metadata visualization for digital libraries: interactive timeline editing and review. Proc. ACM Digital Libraries, pp. 126-133, 1998.

[15] J.P. Freeman, B. McChesney, M. Denari, and S. Thompson. The Explosion in Intelligent Video, International Security Conference West, http://www.iscwest.com/, 2006.

[16] ObjectVideo. http://www.objectvideo.com/products/

[17] V.A. Petrushin. Mining rare and frequent events in multi-camera surveillance video using self-organizing maps. Proc. Conference on Knowledge Discovery in Data, pp. 794-800, 2005.

[18] P. Pillai, Y. Ke, and J. Campbell. Multi-fidelity storage. Proc. ACM Workshop on Video Surveillance and Sensor Networks, pp. 72-79, 2004.

[19] C. Plaisant, B. Milash, A. Rose, S. Widoff, and B. Shneiderman. LifeLines: visualizing personal histories. Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 221-227, 1996.

[20] I.O. Sebe, J. Hu, S. You, and U. Neumann. 3D video surveillance with Augmented Virtual Environments. Proc. ACM Workshop on Video Surveillance and Sensor Networks, pp. 107-112, 2003.

[21] Y. Taniguchi, A. Akutsu, and Y. Tonomura, PanoramaExcerpts: Extracting and Packing Panoramas for Video Browsing. Proc ACM Multimedia 97, pp. 427-436, 1997.

[22] G. Wang, T.-T. Wong, and P.-A. Heng. Real-time surveillance video display with salience. Proc. ACM Workshop on Video Surveillance and Sensor Networks, pp. 37-44, 2005.

[23] J. Wang, M.S. Kankanhalli, W. Yan, and R. Jain. Experiential Sampling for video surveillance. Proc. ACM Workshop on Video Surveillance and Sensor Networks, pp. 77-86, 2003.

[24] G. Wu, Y. Wu, L. Jiao, Y.-F. Wang, and E.Y. Chang. Multi-camera spatio-temporal fusion and biased sequence-data learning for security surveillance. Proc. ACM Multimedia, pp. 528-538, 2003.

[25] B.-L. Yeo and M.M. Yeung, Classification, Simplification and Dynamic Visualization of Scene Transition Graphs for Video Browsing. Proc. IS&T/SPIE Electronic Imaging '98: Storage and Retrieval for Image and Video Databases VI, 1998.

[26] M.M. Yeung and B.-L. Yeo, Video Visualization of Compact Presentation and Fast Browsing of Pictorial Content. IEEE Transactions on Circuits and Systems for Video Technology, Vol. 7(5), pp. 771-785, 1997.

[27] Z. Zivkovic. Improved adaptive Gaussian mixture model for background subtraction. International Conference Pattern Recognition, Vol. 2, pp. 28-31, 2004.