

Interactive Exploration of Surveillance Video through Action Shot Summarization and Trajectory Visualization

Amir H. Meghdadi, *Member, IEEE*, and Pourang Irani

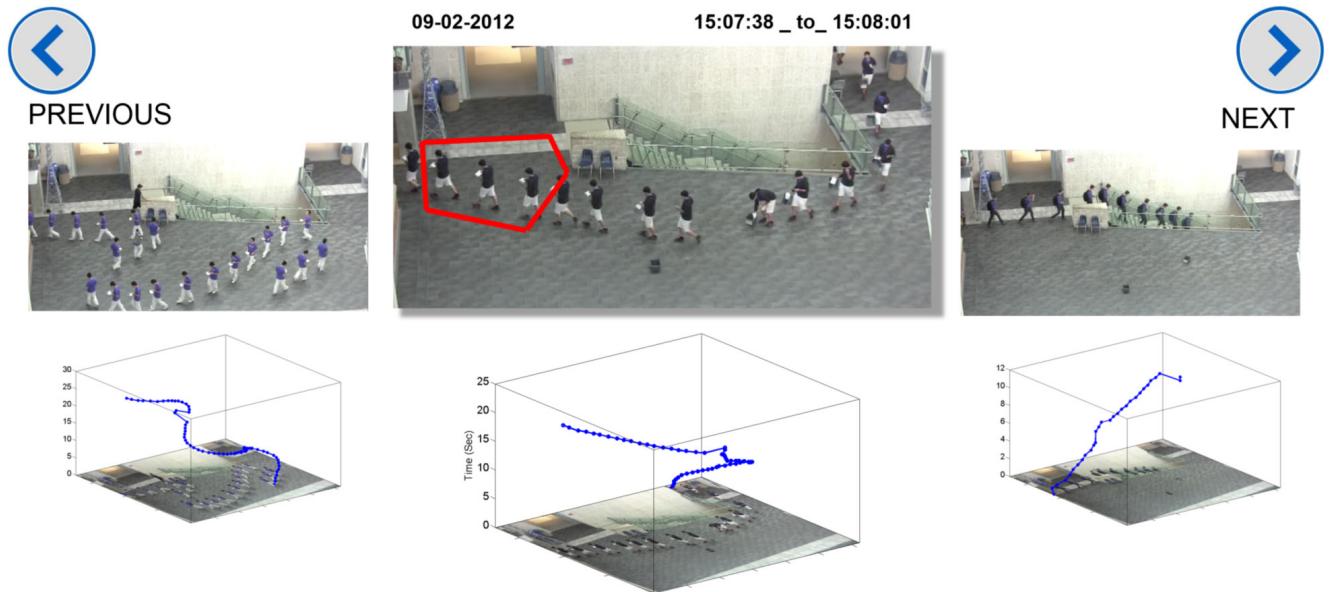


Fig. 1. Our proposed system (sViSiT) extracts motion paths of moving objects in surveillance video. The content of each movement path is made available through action shot images (top images) and properties of the motion itself are visualized using a space-time cube (bottom-images). Users can quickly navigate through such content and its associated trajectories to locate items of interest. Notice that in the middle action shot image, the user can locate the suspect person who left a box on the floor.

Abstract—We propose a novel video visual analytics system for interactive exploration of surveillance video data. Our approach consists of providing analysts with various views of information related to moving objects in a video. To do this we first extract each object's movement path. We visualize each movement by (a) creating a single action shot image (a still image that coalesces multiple frames), (b) plotting its trajectory in a space-time cube and (c) displaying an overall timeline view of all the movements. The action shots provide a still view of the moving object while the path view presents movement properties such as speed and location. We also provide tools for spatial and temporal filtering based on regions of interest. This allows analysts to filter out large amounts of movement activities while the action shot representation summarizes the content of each movement. We incorporated this multi-part visual representation of moving objects in sViSiT, a tool to facilitate browsing through the video content by interactive querying and retrieval of data. Based on our interaction with security personnel who routinely interact with surveillance video data, we identified some of the most common tasks performed. This resulted in designing a user study to measure time-to-completion of the various tasks. These generally required searching for specific events of interest (targets) in videos. Fourteen different tasks were designed and a total of 120 min of surveillance video were recorded (indoor and outdoor locations recording movements of people and vehicles). The time-to-completion of these tasks were compared against a manual fast forward video browsing guided with movement detection. We demonstrate how our system can facilitate lengthy video exploration and significantly reduce browsing time to find events of interest. Reports from expert users identify positive aspects of our approach which we summarize in our recommendations for future video visual analytics systems.

Index Terms—Video visual analytics, surveillance video, video visualization, video summarization, video browsing and exploration

1 INTRODUCTION

Surveillance cameras are ubiquitous and appear in environments such as hospitals, schools or banks. These cameras record an entire day's length of activities resulting in very long video footages that makes the process of browsing video content a laborious and time consuming task for a human observer. Fully automated video analysis methods eliminate the need for a human observer and rely on computer vision and machine learning to detect events of interest. However, these methods are not fully reliable particularly when the search criteria are subjective or vaguely defined.

• Amir H. Meghdadi is with the Department of Computer Science, University of Manitoba, Canada. E-mail: amir@cs.umanitoba.ca.

• Pourang Irani is with the Department of Computer Science, University of Manitoba, Canada. E-mail: irani@cs.umanitoba.ca.

Manuscript received 31 March 2013; accepted 1 August 2013; posted online 13 October 2013; mailed on 4 October 2013.

For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

Researchers have proposed various solutions to this problem by summarizing videos or visualizing the content in various forms that can potentially lead to more efficient manual browsing and exploration. Such human-in-the-loop systems rely on computer vision algorithms that are best suited for processing the video data at lower semantic levels with interaction and visualization features for facilitating robust human judgment at a higher semantic level. For example, detecting a moving vehicle and speculating whether the driver is impaired is a task with both lower and higher semantic elements, respectively. This process, also referred to as *video visual analytics* [13] heavily relies on the interaction between the human operator and the computer system. Video visual analytics systems include a variety of dimensions consisting of video summarization [20][18][29], video content visualization [4][7][11][26] or video interaction and navigation [8][11][12][22][32].

While a few systems have combined various video visual analytics features [13][16], researchers have suggested that further evaluations are necessary to explore how such tools facilitate a variety of analytic tasks [13][16]. Ideally, novel video analytic systems with one or more dimensions can be effective in reducing users' browsing time in a wide array of search and exploration tasks with video data.

In this paper, we present sViSIT (selective Video Summarization and Interaction Tool), a video analytic system that summarizes and visualizes both the video content and the trajectories corresponding to individual moving objects. sViSIT facilitates user interaction through visual inspection of the video snippets and spatio-temporal filtering of the trajectories. We generate action shot images (stroboscopic representations of a moving object in a single image, e.g. see Fig. 1, top) to summarize and visualize the content of each and every movement in a video. We also augment the action-shot images with a space-time cube to visualize the spatial and temporal characteristics of movement trajectories (e.g. see Fig. 2, where the 2D trajectory would be self-intersecting and the 3D space time cube provides a better visualization). The system allows users to select regions of interest and filter the events based on the spatio-temporal characteristics of the movements.

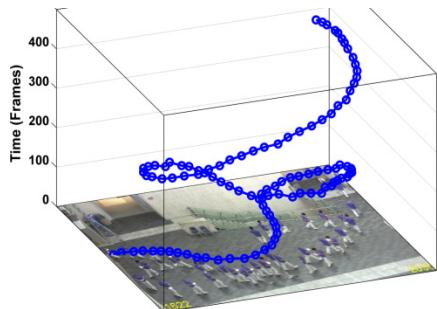


Fig. 2. Space time cube allows spatio-temporal visualization of a movement trajectory

sViSIT enables queries that (a) have subjective search criteria and are therefore challenging for an automatic vision algorithm, and (b) need acute inspection from an operator, such as: “*finding a young man who has entered from a given door and has been loitering around the window*”, or “*finding if and when a large van stopped or suspiciously slowed down on the bridge*”. Such complex tasks are possible primarily using sViSIT’s spatio-temporal querying abilities and action-shot object movement visualization.

Our proposed contributions consist of the following: 1) a novel video visual analytics system (sViSIT) for interactive and analytic exploration of video data. The system links visualizations that contain information about both the specific content (action-shot image) and properties of the movement trajectory (space-time cube) of moving objects. 2) an evaluation consisting of a large variety of analytic tasks derived from our

interaction with professional security personnel who work with such systems on a daily basis.

2 RELATED WORK

Most closely related to our contribution is work in *video visual analytics* [13][16]. We also briefly review work specific to (a) *video summarization*, (b) *video visualization* and (c) *interaction and navigation techniques for video analytics*. Table 1 summarize some of this prior work while a thorough survey of all these elements are available in [3].

2.1 Video Visual Analytics

In [13],[16], the authors present a framework for video visual analytics based on the *VideoPerpetuGram* (VPG) [4] to visualize objects’ movement tracks. VPG is a visualization method that plots abstract illustrations of the movement trajectories along with some key frames at sparse locations. Users can interact with the system by applying filters on the trajectories. In another work [19], movement trajectories have been extracted and displayed on top of key frames from the video, however with limited user interaction. In [14] and [17], a method is presented for clustering, schematic visualization and scatter/gather browsing of the trajectories. Trajectories of moving objects provide useful information about the events in a surveillance video [25]. In all of the above approaches, visualization is centred on the trajectory of the moving objects and the object itself is not visualized. Therefore, these methods are suitable for queries based on the movement characteristics of objects and not on the object itself. However, in many search and exploration tasks the visual appearance of the objects during their movement is as important as their trajectories.

Furthermore, the above systems have provided limited knowledge on browsing and exploration efficiency with multiple tasks. The most closely related user study is conducted in [15], to evaluate the efficiency of various methods for visualizing video frames in fast forward video playback. However, the evaluation is limited and elements of user interaction and movement object path extraction were not tested. To the best of our knowledge, our method of extracting movement paths and showing both movement content and properties is the first to evaluate varied tasks and analytic performance in terms of browsing time efficiency.

2.2 Summarization

Video summarization methods [29][20][18] aim at removing redundancies and providing a shorter, more concise representation of the video data for faster exploration by the human analyst. Fast forwarding and adaptive fast forwarding [15] remove frames and shorten the length of the video. Detecting and displaying key frames only keeps key frames that contain important data. Mosaic images [18] display a larger panoramic view of the scene created by alignment and integration of the frames for a moving camera that shows only the static background and not the moving objects. However, more recent methods in panoramic summary image [2] generate a summary of both the moving object and the background in one single image while the camera is moving. These methods however are better suited for applications in arts, sports and entertainment and not for video surveillance applications where cameras record very long videos.

Methods in summarizing long video footage suitable for video surveillance include Non-chronological Video Synopsis that collapses the spatio-temporal information on the time domain [25] to allow for the simultaneous display of events that happen at different moments. Similarly, Video Montage [19], collapses the data on both the time and spatial domain by pulling segments of the video from different spatial and temporal regions and stitching them together. These methods remove redundancies to display as much visual information as possible in a synopsis video which makes the summarization process very effective. However, visualization in the

Table 1. Selected Previous Work in Video Summarization, Visualization and Interaction

First Author. Year	Method	Contribution	Surveillance Application	Summarization in time domain	Interaction content querying	Object Visualization	Trajectory Visualization
Irani.1996	Mosaic images [18]	Panoramic image representing background contents, moving camera	-	✓	-	-	-
Bartoli.2004	Motion panorama [2]	Panoramic image representing a moving object, moving camera	-	✓	-	✓	-
Kapler.2004	Video Summagator [26]	Visualizes the video cube in 3D and highlights the movements	-	-	-	✓	-
Kang.2006	Video Montage [20]	Collapsing spatio-temporal data in both time and spatial domain.	-	✓	-	-	-
Caspi.2006	Dynamic Stills [5]	Action shots (a.k.a. dynamic stills) visual summaries of a single movement.	-	✓	-	✓	-
Pritch.2008	Video synopsis [29]	Non-chronological synopsis shifts the movements only in the time domain	✓	✓	-	-	✓
Tang.2009	Slit-tears [32]	Visualizing 1D regions of interests (line segments) in a video	-	-	✓	✓	-
Höferlin.2009	Visual analytics [13]	VideoPerpetuGram (VPG) visualization and filtering of trajectories.	✓	-	✓	-	✓
Correa.2010	Video Narratives [6]	An interactive authoring system to create compact representation of the movements using mosaic images	-	✓	-	✓	-
Höferlin.2011	Schematic Summaries[14]	Interactive clustering and schematic visualization of the movement tracks.	✓	✓	✓	-	✓
Sunkavalli.2012	Video Snapshots [31]	High quality still images (snapshots or action shots) from a short video clip	-	✓	-	✓	-
Present paper	sViSIT	Action shot and trajectory visualization in an interactive system	✓	✓	✓	✓	✓

form of synopsis videos is condensed with many moving objects that make the browsing task difficult. In addition, the important information about the timing of events would be lost because they have been shifted in the time domain. These methods do not necessarily provide advanced filtering or content interaction [19][27].

2.3 Visualization

Traditional visualization of video data is limited to displaying the video frames with original or fast forward playback speed. Alternative visualizations have been proposed in recent years. Some of these methods visualize the video cube in 3D in its original [10] or modified shape [7]. Video Summagator [26] provides a 3D visualization of the video cube and highlights the moving objects by blurring the background. Other methods extract and visualize only important information such as movement trajectories [14][17]. Advanced visualization methods often entail summarization as well (described above). Dynamic Stills [5], and Video Narratives [6] for example, both visualize a summary of the video by generating still images or short video clips that show the moving object at various positions in a single image.

2.4 Interaction and Navigation

Interaction with the video content is an important element in video visual analytics as it ensures an interactive exploration of the content. The traditional interaction modalities with videos include *Play*; in normal or modified (e.g. *Fast-Forward*) playback speed, *Pause*, *Jumping* back and forward and *Seeking* using markers on a timeline view to navigate through video content [24]. However, novel navigation techniques have been proposed for content dependent manipulation of the video playback [8][22], or content querying [32]. Slit-tears [32] allows for spatial querying of the video content on 1D regions of interest (line segments) and provides a time line visualization of the pixel variations across frames on the given line segments. However, it does not summarize the video in the time dimension and regions of interests are limited to the 1D area queried.

3 SYSTEM ARCHITECTURE

The architecture of the user interaction is shown in a flowchart in Fig. 3. Automatic video processing and tracking does not require user intervention. The user interaction starts with selecting a region of interest. We describe the algorithmic steps in our visual analytics system as follows:

3.1 Motion Detection and Tracking

As the first step, our motion detection and tracking algorithm detects the moving objects in each video frame and extracts the movement trajectories. We first detect the background and update it every 5 seconds using a temporal median filter applied to the last minute of the video data before each frame (~375 frames at 6.25 frames per seconds), practically enough to converge to the background. We then generate the foreground by taking the absolute difference between each frame and the background at each pixel location.

Fig. 4(a) and (b) show a sample frame and the result of background subtraction (original foreground), respectively. We convert the foreground into a binary (black and white image) mask by thresholding the pixel intensities after noise removal Fig. 4(c). We then use a *closing* morphological operation to close the gaps and generate final blobs (segments) in the binary image. A *connected component detection* algorithm detects and labels each segment to be used as a mask to extract the moving objects Fig. 4(d).

We track the objects by linking each moving object (segment) in each frame to the most similar (having the least distance) segment in the next frame. We define a distance measure between each pair of image segments by considering each segment as a set of points in the feature space of color components. *Mahalanobis* distance is a well-known classical distance function between two data points with respect to distribution of the data points in a feature space [9] and has been used as a reliable distance function between pixel values in RGB color space. The original Mahalanobis distance is defined between a pair of points. However, we propose to use the *generalized Mahalanobis* distance as defined in [1] as the distance function between sets of points.

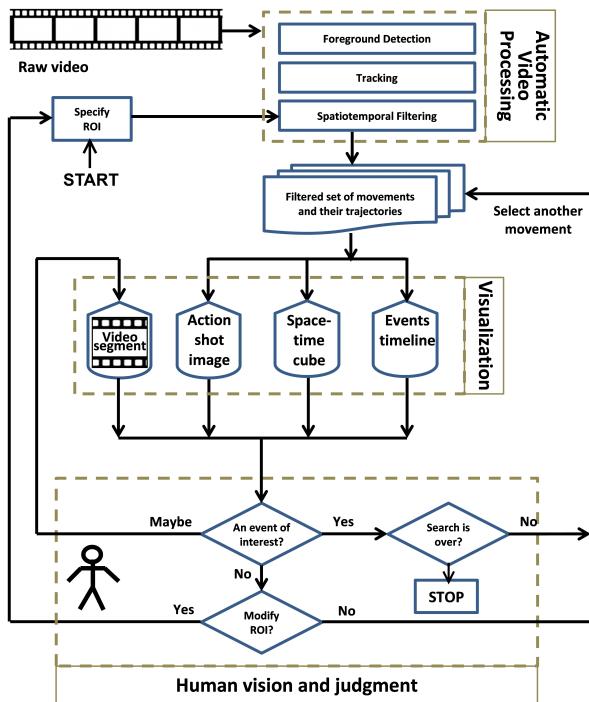


Fig. 3. Flowchart of the proposed system: sViSiT

Let X and Y be sets of feature vectors of pixels corresponding to image segments X and Y in frames F and $F+1$, respectively. The generalized Mahalanobis distance $gMD(X, Y)$ between X and Y is defined [1] as:

$$gMD(X, Y) = \sqrt{(\vec{m}_X - \vec{m}_Y)^T \Sigma_W^{-1} (\vec{m}_X - \vec{m}_Y)}$$

Where \vec{m}_X and \vec{m}_Y are average feature vectors for all the pixels in X and Y and Σ_W is within-class covariance matrix defined in

$$\Sigma_W = \frac{1}{2} \left(\sum_{x \in X} \frac{(x - \vec{m}_X)(x - \vec{m}_X)^T}{n_X} + \sum_{y \in Y} \frac{(y - \vec{m}_Y)(y - \vec{m}_Y)^T}{n_Y} \right),$$

where n_X and n_Y are cardinalities of sets X and Y .

In order to track all objects and extract the movement trajectories in a systematic way, we use a layered graph to represent the moving objects in the video. A layered graph is a graph in which nodes are partitioned into layers L_1, \dots, L_n and all edges are between adjacent layers [28]. Each layer in the graph represents one video frame and contains several nodes that represent image segments (moving objects) in that frame (Fig. 5).

We connect (link) each node in each layer (video frame) to the node in the previous frame that has a minimum distance as explained above and also below a set threshold. We introduce a *depth-first* algorithm to search and detect every possible path (corresponding to a movement trajectory) in the graph. We then use this information to extract movement trajectories and generate action shot summaries (next section). We implemented the system using the above method. More sophisticated tracking algorithms such as [27] can also be used if one requires more robust tracking.

3.2 Visualization and User Interaction

By extracting trajectories, our system provides interactive filtering and retrieval of specific events based on spatio-temporal characteristics of the movements. Users start by selecting region(s) of interest and the system automatically applies the filter and keeps only those movement events where the movement track (trajectory) intersects with the region of interest. Logical combinations (AND/OR) are also possible when dealing with more than one region of interest. A timeline overview of the movement events and the

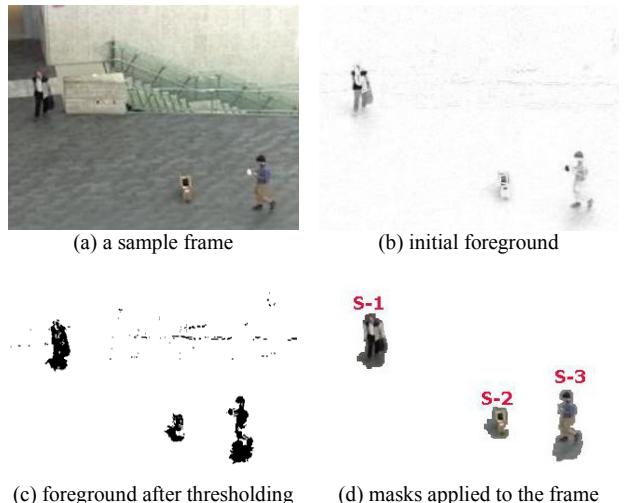


Fig. 4. Foreground detection for a sample frame

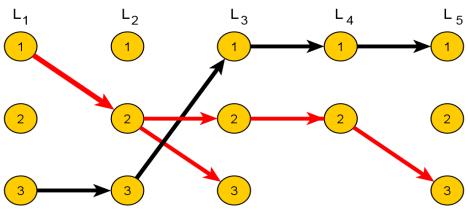


Fig. 5. A layered graph representation of the moving objects in video frames. Each node represents a detected blob in one frame. The red trajectory represents a split in the tracking.

number of retrieved movements is displayed. The user can select any movement for further examination either by a linear search through all of them or by clicking on the corresponding timeline view.

3.2.1 Action Shot Image

For each movement event, we generate an action shot image by extracting the moving object (a segment) in each frame and adding them to the background image to create the stroboscopic effect. To minimize self-occlusion (overlapping the image segments in two consecutive frames), a new extracted segment in each frame is not added to the background unless the overlap between the new segment and the previously added one is smaller than a threshold (e.g. 0 for complete separation). This makes the method less susceptible to changes in the speed of moving objects and independent of frame rate. The user can then browse through action shot images and also play the video segment for each movement to examine the course of events in more detail.

3.2.2 Space-Time Cube

A space-time cube, (e.g. see Fig. 2 and Fig. 8) is a 3D geometrical graph where the base of the cube represents the x-y spatial dimension (image domain) and the cube's height represents the temporal dimension (time domain) [21]. The space-time cube complements the action shot view as it can visualize spatio-temporal characteristics of the movements (such as speed; a steeper line indicates a slower movement) that are otherwise not apparent in the action shot image. Fig. 8 for example shows an action shot that only displays a person approaching a chair while its trajectory in the space time cube clearly shows that the person sat on the chair for more than 30 seconds. Multiple trajectories can be visualized in our system simultaneously and the trajectories can be explored by viewpoint rotation.

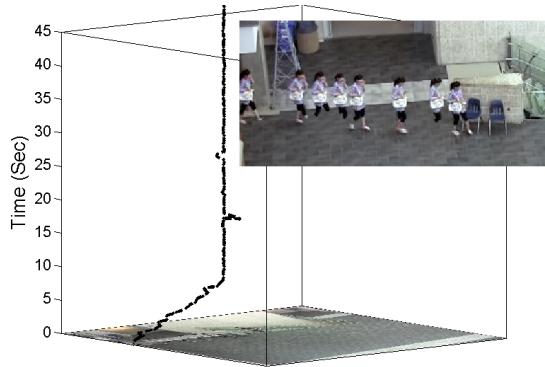


Fig. 6. An example of the movement trajectory in space-time cube (centre) and the action shot image (top right)

4 SYSTEM IMPLEMENTATION

A snapshot of the sViSIT user interface is shown in Fig. 9 (a). We implemented the system in MATLAB® and used the *Image Processing Toolbox* for basic image processing operations such as filtering, morphological closing or connected component detection. Our system is able to work on video segments (we refer to them as *episodes*) extracted from the unlimited stream of the surveillance video data. The user can also move a sliding window to select the time interval of analysis as a *period of interest (POI)* with respect to each query. However, in our user study, we used episodes of half an hour length and used the whole episode as the period of interest to analyse our 30 min long videos at once.

The period of interest (POI) and a bar chart of the number of moving objects in each frame during an episode are plotted (Fig. 9a, part D). An event timeline of all movement events in the period of interest is also generated and displayed (Fig. 9a, part C). The event timeline represents time and duration of each movement event using horizontal bars. The selected action shot and its movement trajectory in the space-time cube are shown in part A and B (Fig. 9a), respectively. Users either browse through all the detected movements using the right and left arrow keys or jump to any part of the video by clicking on the corresponding horizontal bar on the events timelines. The original video segment corresponding to each action shot image can be played in the same window as the action shot image.

In the main user study, we compared our system against manual browsing with fast forward video playing. However, in order to properly evaluate our system and demonstrate the effect of our visualization methods, we implemented a new video player with motion detection capability in regions of interest but without the advanced visualizations in sViSIT. We refer to this video player as FFMD: Fast Forward Motion Detection. Fig. 8 shows the controls in the FFMD user interface. FFMD has a slider as well as a timeline view that displays temporal regions in a video where moving objects are detected in selected region(s) of interest. Users can navigate the video through *Play*, *Pause* and *Fast-Forward* playback (with varying speeds). Also, they can *Seek* movement activities by clicking on the motion detection timeline to jump into part of the video (highlighted with green bars) when moving objects are detected inside the regions of interest. This interface is akin to those currently available on commercial systems (*e.g.* Panasonic® [30]), in terms of functionality (*i.e.* motion detection in regions of interest and fast forward playback).

5 SYSTEM EVALUATION AND USER STUDY

We conducted a two part evaluation of our system. One of the primary concerns of our domain experts, campus security personnel,

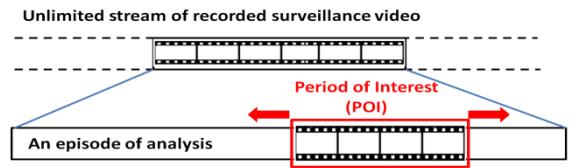


Fig. 7. An episode of video data



Fig. 8. Vertical bars on a timeline in FFMD player represent the presence of an object in the region of interest. Users can click on the bars and it will jump to that location in the video.

consisted of improving and making the analysis more efficient. An initial controlled experiment allowed us to identify the value of sViSIT for efficient video browsing. We also met with domain experts to receive feedback on how well our design meets their analytic requirements. We hypothesized that performing common video analytic tasks will be more efficient in sViSIT in comparison to FFMD, a user friendly implementation of the common method for browsing videos. We also hypothesized that performance with sViSIT is an improvement over the general fast-forward approach. We chose a 5x playback speed as a baseline as in [22] it was shown that the ability to understand the content of the video would start degrading if the playback speed increased more than about 5 times the original speed. Our domain experts also indicated that this was the typical fast-forward speed they would use.

We expect the time-to-completion of a task (TTC) to depend on when an event (or target) appeared in the video relative to its start (we refer to it as T_t). We also hypothesize that there are various other factors that could affect the performance of the search and exploration such as the type of search criteria, how crowded the frames are in the video and the size of the user selected region of interest (ROI). We investigate the effect of these factors on analytic performance.

5.1 Materials

We recorded 120 minutes of video data (4 different videos, 30 min each) from various indoor and outdoor public places. The content included movements of people and vehicles (Fig. 9, b.1 to b.4). We down-sampled the video frame size to 608×336 pixels and video frame rate to 6.25 frames per second to make it appear similar to common video surveillance data.

5.2 Tasks

In consultation with our selected domain experts (campus security) we identified 14 different tasks that require browsing the videos in search of specific events (targets) of interest (Table 2). We evaluated our system by measuring the time-to-completion of the above tasks. In our tasks, targets appear at different times in the videos as shown in Table 2. This will prevent users from making any assumption about the temporal locations of the targets and to prevent biasing the results. To minimize the selection bias, we randomly selected our tasks and videos before the user study and included all the results. Our tasks can be categorized based on the task type, video crowdedness and size of the region of interest. The various task type categories include:



(a) sViSIT: selected action shot image (A), the space time cube (B), the events timeline in the adjustable period of interest marked between red and blue lines (C), timeline of the full episode displaying the number of moving objects in each and every frame (D). Events during the period of interest are visualized.

Fig. 9. (a) a snapshot of our tool (sViSIT), (b1)-(b4) one sample video frame and the region of interest from each video in the user study

Type A) *Presence in ROI*: Detecting the presence/duration of a specific type of object (e.g. cars driving in a street, pedestrians crossing, objects left behind ...). Experts suggested that this was one of the most common tasks they performed to detect loitering, vandalism and other similar events based on a specific location.

Type B) *Activity detection*: Detection of a specific activity (e.g. handshaking, talking, vandalism ...). This was a frequent task reported by our domain experts and one which involved identifying specific actions even after localizing the time period in which the event occurred.

Type C) *Visual search for a given target*: Detection of a specific given target that is shown to the user in an image (e.g. tasks 5,9,12). This task category was performed less frequently by our experts but occurs when witnesses identify a potential suspect.

We also defined three levels for the size of the region of interest. These levels are: VL (*Very Limited*: a small part of the video is selected as the region of interest), L (*Limited*: part of the video frame is selected as the region of interest) and NL (*Not limited*: the region of interest is very large or non-existent, i.e. the entire video viewing angle). The region of interest limits the search of movement paths to those that intersect with the region.

We also categorized our videos based on the level of movement activities. We call these categories *Uncrowded (UC)*, *Crowded (CR)* and *Very Crowded (VC)*, wherein the first signifies very little movement and the latter represents a high amount of movement. Most often, according to our domain experts, events of a critical nature take place in the former or uncrowded activity areas.

We first conducted a pilot study with 8 participants who performed all the above tasks using sViSIT. The purpose of this pilot study was to test the system performance by itself, the feasibility of the user study for the designed tasks and user feedback about the

tool. In our feedback forms, 7 participants described the system as *Very Useful* and one participant as *Useful*.

5.3 Subjects

For our controlled experiment, we recruited 18 volunteer participants between the ages of 21 and 32 (university students with no prior knowledge about this project, not including the participants in the pilot study). 17 were male. We asked them to perform the 14 tasks in the same order listed in Table 2. None were experts in video analysis.

5.4 Procedure

All sessions were supervised to ensure that users were exploiting all the features of the tool. Each session started with 15 min of introduction and training and lasted about one hour in total for each participant. Subjects sat directly in front of the monitor at a distance of about 60-70 cm and the size of the videos/images displayed was 21×12 cm on both sViSIT and FFMD user interfaces. The participants were randomly divided into two groups to counter balance the assignment of the tools to task numbers. The first (second) group performed odd (even) numbered tasks with sViSIT and the remaining tasks with FFMD. Each task was delivered to the users through a set of instructional dialog boxes which asked the participants to read and understand the task description before they proceeded with the task. The task descriptions and the target images to be shown to the participants (type C) were displayed on a separate monitor to their left side. Participants were instructed to click on a STOP button as soon as they found the target at which point the time-to-completion was recorded and they were able to write down the answers in a dialog box. In one trial, the user gave up after searching in the video for 12 min using the FFMD player and this trial data was discarded. We presented the tasks one by one in the given order and we supervised the study to make sure users did not gain prior knowledge about the tasks.

Table 2. Task List, Target Time (T_t) and Average Time-to-Completion TTC for sViSIT and FFMD

Task Vid	Task description	Type	Crowd edness	ROI	T_t mm:ss	Avg. TTC sViSIT	Avg. TTC FFMD
1 2	Find a cyclist riding along the street.	A	UC	L	8:42	00:31	00:49
2 2	Find a pedestrian crossing the street.	A	UC	L	3:19	00:39	01:16
3 2	Find a pedestrian walking in the parking lot.	A	UC	L	5:37	00:23	00:31
4 2	Count the number of vehicles driving along the street.	A	UC	L	12:24	00:30	01:22
5 4	Find the person shown in Fig. 10(a) (carrying a green bag).	C	CR	L	18:50	01:41	02:31
6 4	Find the person in red jacket entering the scene from a given corner.	A	CR	VL	19:54	01:14	01:52
7 4	Find two people who meet and handshake.	B	CR	NL	10:46	02:23	01:13
8 4	Find the suspicious person who talks to someone on the roof	B	CR	NL	2:13	00:38	00:52
9 3	Find the person shown in Fig. 10(b) (carrying plastic boxes)	C	VC	NL	18:16	03:50	02:34
10 3	Find a box being dropped off on the ground in a given area.	A	VC	VL	28:08	02:01	03:47
11 3	Find the person who commits vandalism on the given poster.	B	VC	VL	28:16	00:47	01:52
12 3	Find the person shown in Fig. 10(c) (carrying boxes)	C	VC	NL	29:39	01:50	03:58
13 1	Find the person who is loitering around the entrance door.	A	UC	VL	9:20	00:53	00:34
14 1	Find the person who leaves a recycle bin on the top floor Fig. 10(d)	C	UC	L	11:31	00:52	01:21

Legend:
 UC(Uncrowded),
 CR(Crowded),
 VC(Very crowded),
 NL(Not limited),
 L(Limited),
 VL(Very limited)



Fig. 10. Target images presented to the users in type C tasks (a) to (d)



Fig. 11. Target events in type A and B, not presented to the user, (e) to (m)

5.5 Results and Discussion

Table 2 shows average time-to-completion (TTC) of the tasks for sViSIT and FFMD video player (last two columns). TTC values are considerably smaller than the target appearance times T_t in the video, which demonstrates the advantage of using our tools compared to manual browsing of the video with original speed. As expected, TTC is highly correlated with the relative target appearance time (T_t) in the video, *i.e.* a target toward the end of the video will take much longer to find than one toward the start. Therefore, we define and use a normalized time-to-completion $nTTC = TTC/T_t$ to be able to compare the time-to-completion between different tasks independent of T_t . Fig. 12(a) shows the average and standard error of the normalized time-to-completion ($nTTC$) plotted for sViSIT and FFMD for all tasks sorted based on T_t .

We also compare $nTTC$ with a baseline value of 0.2 which corresponds to the target appearance time in 5x fast forward video playback. According to Fig. 12(a), targets that appear in the video too early (less than about 5 min, $T_t < 5:00$, 1 min in a 5x fast-forward), are more likely to be detected faster by watching the 5x

fast forward video. This is expected because it always takes some time to select regions of interest and interact with the video.

5.5.1 The Effect of the Browsing Tool, Task and Video Type

One way ANOVA shows the significance of the effect of the following factors on average normalized time-to-completion ($nTTC$) as follows.

The tool (sViSIT versus FFMD): overall, average performance ($nTTC$) of sViSIT and FFMD are significantly different (Fig. 12b). sViSIT is more efficient than FFMD ($F(1,17)=5.584$, $p=0.03$).

Search type (A, B or C): there was a significant difference between tasks of type C (visual search for a given target) and other types in favour of type C (Fig. 13b)

Crowdedness: We did not find any effect on $nTTC$ due to crowdedness factor in our 4 videos.

Limitedness of the region of interest: There is a significant difference between Limited (L)/Very limited (VL) and Not limited (NL) regions of interest. More limited regions of interest result in less $nTTC$ as expected. (Fig. 13a)

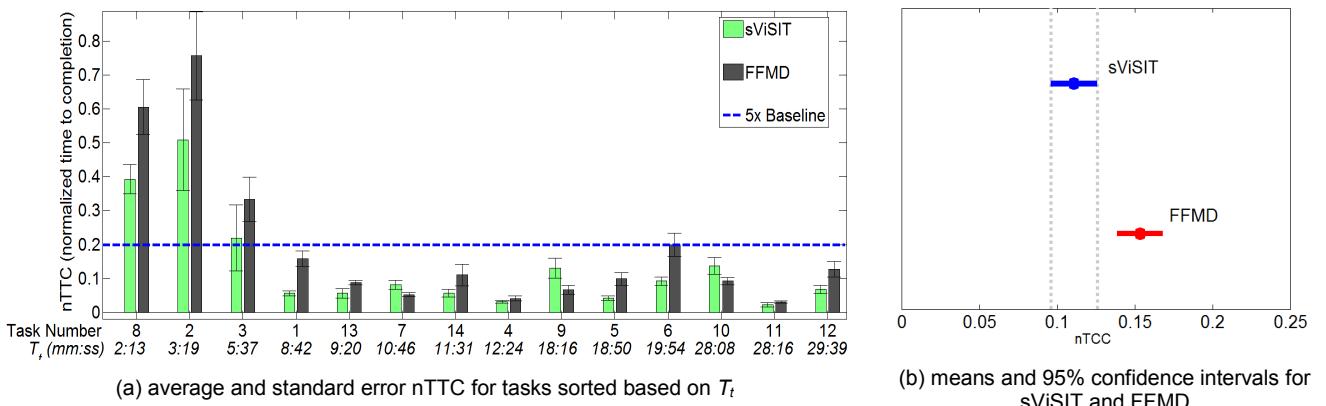


Fig. 12. Normalized time-to-completion (nTTC) differences between sViSIT and FFMD, (a) average nTTC plotted for each task compared to baseline value of 0.2 (5x fast forward), (b) comparison between estimated means of trials in sViSIT and FFMD group

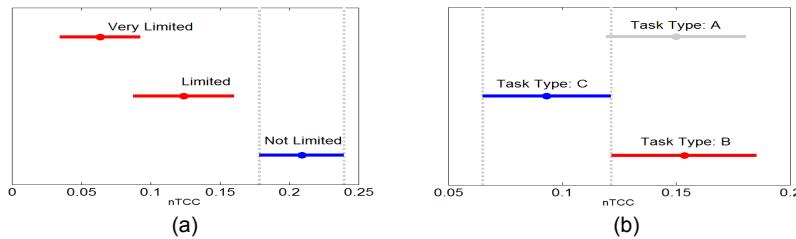


Fig. 13. Overall nTTC differences due to limitedness of ROI and task type (Crowdedness has no visible effect on nTTC)

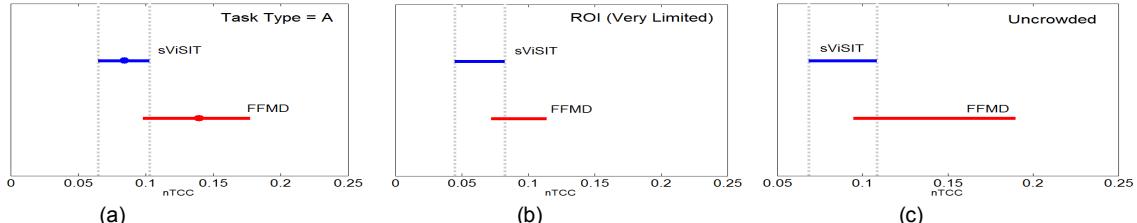


Fig. 14. Significant nTTC differences between sViSIT and FFMD (%95 confidence intervals) for each condition (a) Task type = A, (b) ROI = Very Limited and (c) video is Uncrowded

We did not find significant differences when considering the interactions between the above factors in a multi-way ANOVA. However, we found the following (Fig. 14) for one way analysis of variance of nTTC for sViSIT and FFMD (sViSIT performs better on average in all cases). sViSIT is more efficient than FFMD, for task type A ($F(2,34)=5.967, p=0.006$), when the video is *Uncrowded* ($F(2,34)=4.55, p=0.017$) or if the ROI is *Very Limited* ($F(2,34)=4.46, p=0.019$).

5.6 Subjective Evaluation and Discussions

Post-experiment, participants completed a multiple choice questionnaire to provide subjective feedback about sViSIT and FFMD. 88% and 12% indicated that sViSIT is “*much better*” and “*better*” respectively. The other choices (which were not selected by any participant) were “*FFMD much better*”, “*FFMD better*”, “*both are equally good*” and “*none of them are good*”. This shows the overall impression of the users was strongly in favour of sViSIT although on average trends show that they performed three out of 14 tasks faster with FFMD.

We watched and examined the user behaviour while participants performed the tasks. Our qualitative observations indicate that the performance also depends on the user strategy (especially for selecting the region of interest) and the level of experience. For example, some users got frustrated with the high number of detected movements that they needed to browse and thus kept changing the

ROI. This increased their search times, but also refined their ROIs. Other users kept browsing through the results and found the target after the first set of results were obtained based on ROI selection. Some users showed innovative ways of selecting the region of interest. Fig. 15, for example shows an ROI selected by one of the users to capture movements in all directions including the target image shown. We expect that with more experience, users will be able to include varying levels of selection to improve their performance. Our qualitative observation of the user study indicates that the main advantage of sViSIT over video playback is the ability of filtering (through selection of ROI) and summarizing (through action shot images and space time cube). We also noticed that for the given tasks in our user study, users relied more on action shot images rather than space time cubes. This is consistent with a feedback from



Fig. 15. An example of an interesting ROI selected by a user

our domain experts (section 5.7 below) who believe there is more training needed for the space time cube. However, if the tasks require detection of the speed of moving objects (*e.g.* finding a car that slows down in a highway), then we expect use of the space time cube to be inevitable as other visualizations do not provide information about the speed of moving objects. The ability to visualize trajectories of more than one moving object in the space time cube is another useful feature for our space time cube.

5.7 Expert Evaluation

We also asked our domain experts to evaluate and provide feedback on the components of our system, the action shots, that showed content of movement trajectories, the profiles of trajectories and the ability to select regions and time intervals. One security officer performed the tasks in Table 2 using our tool. He then indicated the benefits offered by our tool and three other members shared their opinions upon observing the user study. They indicated the significant benefits offered by our tool. We include below a few of their comments.

- The events timeline (Fig. 9.C) was seen as being very useful as it can quickly show the events with the longest duration. This could be an indicator that an individual is spending significant time in a given zone. They used the events timeline before browsing and using any other feature in sViSiT.
- They indicated that the space-time cube was slightly difficult initially but was then deemed useful after 15 minutes of experience with the system. For example, they indicated that an emerging pattern during vehicle break-in involves a suspect visiting multiple cars in the parking lot. This pattern would be clearly visible as a series of arcs in the space-time cube.
- They found the action shot images to be useful for both analytic purpose and documenting specific events. An action shot depicting a suspect in various positions could provide a visual summary of the activity and deemed this to be highly beneficial in incident reports.
- In many of the search and exploration tasks, an officer needs to locate an incident (*e.g.* vandalism) within a long time-frame (up to 48 hours). The operator follows a *binary search algorithm* to locate the time of incident which occurs when a noticeable change occurs in the scene (*e.g.* a broken window). This strategy could be very time consuming and even impossible if the incident does not leave a noticeable change (for example, task 11, Fig. 11(k) in our user study). sViSiT was shown to be highly useful in locating such incidents.

5.8 Limitations

If a video is too crowded with dense activity and without any specific criteria on the movement trajectory, there is no gain in summarization and filtering. However our system is powerful in video surveillance applications where the task often involves searching for a person/activity in a limited region of interest during a long time-frame (period of interest) with sparse activities. We have implemented a general-purpose detection and tracking algorithm in our video processing stage with no training required and with a small number of limitations (*e.g.* occlusion or merge/split of different moving objects). Therefore, the tracking system is prone to limitations of the current state of the art in the field of tracking. However, such limitations only affect the performance and not reliability of our system because the visualizations are still meaningful for a human observer who makes the judgements and has the ability to examine the raw video and confirm his/her findings.

5.9 Computational Cost

Although our current implementation of sViSiT in MATLAB® is not optimized for speed, it is fast enough for real time motion detection up to 3 frames/sec (608×336 pixels) of the video stream. We recorded the average computational time per video frame for one of the videos (video 4, average crowdedness) as an example. The

average time needed for detection, tracking and creating the visualizations are 137, 93 and 75 milliseconds, respectively, for a total of 305ms equivalent to about 3 frames per second. This means the processing phase can be done in parallel with video recording, storing data for user interaction since the system is designed for offline browsing of the recorded video. The next phase (which is user interaction, querying and spatio-temporal filtering), is executed within a second using the existing data.

6 SUMMARY AND CONCLUSIONS

We presented and evaluated a video visual analytics system (sViSiT) that can help users search a video and find targets in a fast and efficient way. We showed that except for very short videos (less than 5min in our examples), using sViSiT is considerably faster than waiting for the target event to happen in a 5x fast-forward video (Table 2 and the baseline in Fig. 12). Moreover, statistical analysis of the time-to-completion in section 5.5 (Fig. 14) confirmed our hypothesis that sViSiT performs significantly better than FFMD. This superiority is shown to be more substantial in less crowded videos with a long time-frame of interest and limited region of interest. This is expected as the power of our system is due to its ability to summarize movements individually and apply spatio-temporal filters to limit the search results. For our future work, we intend on expanding the querying capabilities of our system based on various characteristics of the movement trajectories such as speed, direction and start/end points. More advanced methods of querying are potentially possible through user-drawn target trajectories.

ACKNOWLEDGMENTS

The authors wish to thank Christopher Bohonis, Tyler Green and Security Services at the University of Manitoba for their input in identifying the important video analytic tasks and for providing feedback on the final deployment of our tools. We also thank anonymous reviewers and our lab members who provided feedback on this manuscript.

REFERENCES

- [1] R. Arai, and S. Watanabe, “A Quantitative Method for Comparing Multi-Agent-Based Simulations in Feature Space,” *Multi-Agent-Based Simulation IX, Lecture Notes in Computer Science*, pp. 154-166, 2009.
- [2] A. Bartoli, N. Dalal, and R. Horaud, “Motion panoramas,” *Computer Animation and Virtual Worlds*, vol. 15, no. 5, pp. 501-517, 2004.
- [3] R. Borgo, M. Chen, B. Daubney, E. Grundy, B. Höferlin, D. Weiskopf and X. Xie “A Survey on Video-based Graphics and Video Visualization,” *Proc. of the EuroGraphics conf., State of the Art Report*, 2011.
- [4] R. Botchen, F. Schick and T. Ertl, “Action-Based Multifield Video Visualization,” *IEEE Trans. Visualization and Computer Graphics*, vol. 14, no. 4, pp. 885-899, 2008.
- [5] Y. Caspi, A. Axelrod, Y. Matsushita and A. Gamlil “Dynamic stills and clip trailers,” *The Visual Computer*, vol. 22, no. 9, pp. 642-652, 2006.
- [6] C. D. Correa and K.-L. Ma, “Dynamic video narratives,” *ACM Trans. Graph.*, vol. 29, no. 4, pp. 88:1-88:9, 2010.
- [7] G. Daniel and M. Chen, “Video Visualization,” *Proc. of IEEE Visualization 2003*, pp. 409-416, 2003.
- [8] P. Dragicevic, G. Ramos, J. Bibliowicz, D. Nowrouzezahrai, R. Balakrishnan and K. Singh, “Video browsing by direct manipulation,” *Proc. SIGCHI Conference on Human Factors in Computing Systems*, pp. 237-246, 2008.
- [9] R. O. Duda, P. E. Hart, and D. G. Stork, “Pattern Classification,” Wiley, New York, 2001.
- [10] S. Fels and M. Kenji, “Interactive video cubism,” *Proc. of 1999 workshop on new paradigms in information visualization and manipulation*, pp. 78-82, 1999.

- [11] D. B. Goldman, B. Curless, D. Salesin and S. M. Seitz. "Schematic storyboarding for video visualization and editing," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 862-871, 2006.
- [12] K. Hasan, T. Grossman and P. Irani, "Comet and target ghost: techniques for selecting moving targets," *Proc. of SIGCHI Conference on Human Factors in Computing Systems*, pp. 839-848, 2011.
- [13] M. Höferlin, B. Höferlin, and D. Weiskopf, "Video visual analytics of tracked moving objects," *Proc. of Workshop on Behaviour Monitoring and Interpretation, BMI* 2009, vol. 541, pp. 59-64, 2009.
- [14] M. Höferlin, B. Höferlin, D. Weiskopf and G. Heidemann, "Interactive schematic summaries for exploration of surveillance video," *Proc. ACM International Conference on Multimedia Retrieval, ACM ICMR 2011*, pp. 9:1-9:8, 2011.
- [15] M. Höferlin, B. Höferlin, D. Weiskopf and G. Heidemann, "Evaluation of Fast-Forward Video Visualization," *IEEE Trans. Visualization and Computer Graphics*, vol. 18, pp. 2095-2103, 2012.
- [16] M. Höferlin, B. Höferlin, D. Weiskopf and G. Heidemann, "Uncertainty-aware video visual analytics of tracked moving objects," *Journal of Spatial Information Science*, no. 2, pp. 87-117, 2011.
- [17] M. Höferlin, B. Höferlin, G. Heidemann and D. Weiskopf "Interactive Schematic Summaries for Faceted Exploration of Surveillance Video," *IEEE Trans. Multimedia*, vol. 15, no.4, pp. 908-920, 2013.
- [18] M. Irani, P. Anandan, J. Bergen, R. Kumar and S. Hsu, "Efficient representations of video sequences and their applications," *Signal Processing: Image Communication*, vol. 8, no. 4, pp. 327-351, 1996.
- [19] J. Zhong, S. Yuting, R. Qian and M. Jintao "Surveillance video summarization based on moving object detection and trajectory extraction," *International Conference on Signal Processing Systems, ICSPS 2010*, pp.V2-250-253, 2010.
- [20] H. Kang, X. Chen, Y. Matsushita and X. Tang, "Space-time video montage," *IEEE Conf. on Computer Vision and Pattern Recognition, CVPR2006*, vol. 2, pp. 1331-1338, 2006.
- [21] T. Kapler and W. Wright, "GeoTime Information Visualization," *IEEE Symp. Information Visualization, INFOVIS'04*, pp. 25-32, 2004.
- [22] D. Kimber, T. Dunnigan, A. Grgensohn, F. Shipman, T. Turner and T. Yang "Trailblazing: Video playback control by direct object manipulation," *IEEE International Conference on Multimedia and Expo*, pp. 1015-1018, 2007.
- [23] K. Kurihara "CinemaGazer: a system for watching videos at very high speed," *Proc. of International Working Conference on Advanced Visual Interfaces*, pp. 108-115, 2012.
- [24] F. Li, A. Gupta, E. Sanocki, L. He and Y. Rui, "Browsing digital video," *Proc. of SIGCHI conference on Human Factors in Computing Systems*, pp. 169-176, 2000.
- [25] B. Morris and M. Trivedi, "A Survey of Vision-Based Trajectory Learning and Analysis for Surveillance," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 18, no. 8, pp. 1114-1127, 2008.
- [26] C. Nguyen, Y. Niu and F. Liu, "Video Summagator: an interface for video summarization and navigation," *Proc. of SIGCHI Conference on Human Factors in Computing Systems*, pp. 647-650, 2012.
- [27] N. Papadakis and A. Bugeau, "Tracking with Occlusions via Graph Cuts," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp.144-157, 2011.
- [28] C. H. Papadimitriou, and Y. Mihalis, "Shortest paths without a map," *Theoretical Computer Science*, vol. 84, no. 1, pp. 127-150, 1991.
- [29] Y. Pritch, A. Rav-Acha, and S. Peleg, "Nonchronological video synopsis and indexing," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp.1971-1984, 2008.
- [30] Panasonic video surveillance products, [online] 2013 <http://www.panasonic.com/business/psna/products-surveillance-monitoring/index.aspx> (Accessed 21 June 2013).
- [31] K. Sunkavalli, N. Joshi, S. Kang M. Cohen and H. Pfister, "Video Snapshots: Creating High-Quality Images from Video Clips," *IEEE Trans. Visualization and Computer Graphics*, vol. 18, pp. 1868-1879, 2012.
- [32] A. Tang, S. Greenberg and S. Fels, "Exploring video streams using slit-tear visualizations," *Extended Abstracts on Human Factors in Computing Systems*, pp. 3509-3510, 2009.
- [33] A. Yilmaz, O. Javed and M. Shah, "Object tracking: A survey", *ACM Computing Surveys*, vol. 38, no. 14, 2006.