19

A Parallel Non-Linear Surveillance Video Synopsis System with Operator Eye-Gaze Input

Ulas Vural and Yusuf Sinan Akgul
GIT Vision Lab, Department of Computer Engineering,
Gebze Institute of Technology
Turkey

1. Introduction

Most living areas such as parks, streets, metro stations, shopping centers, schools and houses are monitored by technologically varied surveillance camera systems. While the first generation primitive systems are still largely used to view and record visual data, the semi-autonomous second generation systems started to emerge to help human operators by processing video data to alert them for abnormal situations (Ahmad et al., 2007). These systems generally include an advanced component for motion detection, object recognition, tracking, behavior understanding, video indexing, or video retrieval (Hampapur et al., 2007). Although these systems intend to handle the events automatically (Hu et al., 2004), fully automated surveillance systems are still inadequate (Keval & Sasse, 2006) and therefore human operators are still indispensable (Siebel & Maybank, 2004).

The number of cameras in surveillance systems is getting larger with the increased demand for security in public spaces (Koskela, 2000). While the number of cameras increases, the amount of data and the amount of visual stimuli for an operator become extremely large. Human operators sometimes have to monitor many video feeds at the same time but the visual limitations of human being give permission to handle only a small subset (Preece et al., 1994). These limitations cause operators to overlook some important actions, requiring more operators to maintain a reliable surveillance system. However, the increased number of operators makes the system more reliable but less efficient. The cost of manpower becomes the dominating factor in the total operational cost and it is generally much larger than the costs of software and storage medium (Dick & Brooks, 2003).

Indispensability of human power in surveillance systems increases the human workload. Performance of human operators become a key point on the total performance of surveillance systems. Attention levels of operators, their monitoring strategies, characters and moods effect the system's success. It is claimed that the performances of human operators are not stable and even expert operators loses their attention after twenty minutes(Green, 1999). While surveillance systems are so common and the operators are the most important part of these systems, many micro-social studies have been conducted for better understanding of security rooms(Keval & Sasse, 2006; Norris & Armstrong, 1999; D., 2004). All these studies, which show that human attention levels have to be monitored regularly.

The field of Human Computer Interaction (HCI) became very interested in analyzing and designing systems for the interaction between the human operators and the surveillance systems. A large amount of work has been conducted on surveillance systems (Ahmad et al., 2007) to achieve higher efficiency and reliability, which can be separated into two groups. The first group generally works at real-time rates while human operators are monitoring the scene. These systems support operators by placing the views of detected threats in conspicuous places (Steiger et al., 2005). Although these systems are generally limited with a fixed number of objects or actions, they successfully decrease the amount of workload where properties of monitored objects or actions are known. An automated surveillance system consists of a number of complex mechanisms according to its objectives (Hu et al., 2004) like tracking pedestrians, making crowd analysis (Siebel & Maybank, 2004), extracting motion patterns (Gryn et al., 2009) and object recognition (López et al., 2006). Some surveillance systems use advanced user interface designs to make themselves convenient and manageable. The efficiency of operators can be increased by utilizing hand gestures for selecting cameras, zooming, tilting and focusing instead of using traditional mouse and keyboard units (Iannizzotto et al., 2005). Although advanced user interfaces and automatic detection of suspicious threats make the operators more efficient on the monitoring process, the operators might still overlook some important actions.

Retrieving a previously overlooked threat is in the scope of the second group. Since the amount of surveillance-video data is very large, manual reexamination of all the recorded data is time consuming even in accelerated modes. The solution is the searching of actions or objects by using image and video understanding methods. Indexing video data and collecting them in databases increase the speed of subsequent searches (Dick & Brooks, 2003). Content-based video retrieval methods can retrieve objects by considering their shape, color or texture properties but cannot successfully determine specified behaviors (Hu et al., 2004). Tracking of actions and spatial positions of objects are also used for detecting anomalies in surveillance videos (Buono & Simeone, 2010). These systems need the knowledge of event locations and can only work after the event occurs. There are also systems that perform semantic analysis of actions in videos for video indexing (Snoek & Worring, 2005). These methods are more advanced than content-based methods but they have to find low-level visual features and handle semantic video indexing.

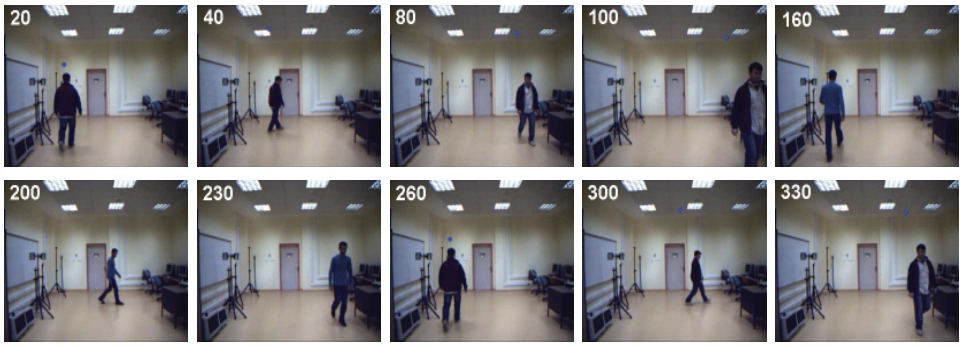
These two groups largely cover almost all the approaches of interactive surveillance systems but there is still a gap between the two groups. Methods in the first group aim to decrease the rate of overlooking but they cannot do anything when operator overlooks suspicious actions. They do not know if the operator perceives the action or not. Methods in the second group support indexing and retrieving of actions. While these methods can be used off-line, they cannot preclude damages of suspicious actions. In addition, actions and their features have to be precisely described to the system. We propose a new eye-gaze based user interface system that can help close this gap. The system neither processes video for the known threats nor indexes actions but it catches the overlooked actions and prepares a summarized video of these actions for later viewing. Our user interface increases the reliability of the surveillance system by giving a second chance to the operator. The system increases the efficiency of operators and decreases the workload by re-showing only a summary of overlooked actions. The system can also be used to summarize video sections where the operator pays most attention. Such a video can be used to review the surveillance video by other operators in a much shorter amount of time.

Our system employs eye-gaze positions to decide operator's Region Of Interest (ROI) on the videos. Eye-gaze based ROIs are used on images for personalized image retrieval and indexing (Jaimes et al., 2001; Jing & Lansun, 2008) but they are not popular on videos. Eye-gaze information is used as a semantic information on images and they cooperate with other content-based methods. While images contain only objects, there are both objects and actions on videos, so finding semantic rules for videos is harder. We do not try to form semantic rules for actions, we only focus on how people watch videos and track motion (Jacob, 1991). Psychological studies show that humans can track only 5 to 8 moving objects at a time (Franconeri et al., 2007; Pylyshyn & Storm, 1988; Sears & Pylyshyn, 2000) by focusing at the center of moving objects instead of making saccades between them (Fehd & Seiffert, 2008). Although expert human operators can track slightly more actions than untrained operators (R. et al., 2004), they may still overlook some important actions at rush times. We propose to estimate video sections that correspond to these overlooked actions by finding video regions with actions away from the center of focus. These estimated video sections are used to produce the final summary video. Similarly, as mentioned before, our system allows video summaries that include only the video sections where the surveillance operator pays attention, which could be used for fast peer reviewing of already monitored videos.

There are many video summarization methods available in the literature (Komlodi & Marchionini, 1998; Li et al., 2009; Truong & Venkatesh, 2007). A classification of linear video summarization methods is given by Li et al. (2001). The most popular video summarization methods are based on discarding frames with least activity (Kim & Hwang, 2000; Li et al., 2000), but this simple method cannot compress a video shorter than number of possible key frames. These methods need a threshold and it is not generally possible to determine this threshold perfectly, lower thresholds increase size of the summarized video and higher thresholds discard the frames with activities.

Another important problem with the methods that discard whole-frames is that the summarized videos might contain both overlooked and focused actions if they are in the same frame. We need a summary method that lets objects move on the time axis independently to compress the activity from different time intervals into a very small time volume. One such method is the non-linear video summarization approach by Acha et al. (2006) who represented the video summary as an energy minimization over the whole video volume. The chronology of a single pixel value is allowed to change, meaning that events of different time steps for the same region of the video image can be collated in any order. In the final summarized video, a single frame is most likely composed of activity from different frames of the original video. For example, for an input video where two persons walk in different frames (Fig 1. (a)), they are seen walking together in its non-linear summary (Fig. 1. (b)). While this method may seem like a good solution for a compact video for later viewing, the overall energy minimization is very complex and it is not suitable for our real-time purposes.

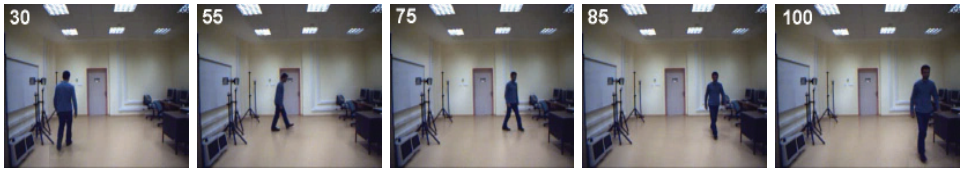
Non-linear video summarization methods are getting popular on surveillance domain (Choudhary & Tiwari, 2008;Slot et al., 2009;Pritch et al., 2009;Chen & Sen, 2008). An extension of 2D seam carving (Avidan & Shamir, 2007) is applied for achieving a content-aware synopsis video for stationary cameras (Slot et al., 2009). A more complex non-linear method using min-cut optimization technique on 3D video volume is proposed by (Chen & Sen, 2008). This method better preserves the actions in a very compact synopsis, it requires large memory space. Another original approach for non-linear synopsis is the grouping of actions(Pritch et al., 2009). Pritch et al. propose an unsupervised system that clusters similar actions. A more



(a) Sample frames from an input video sequence of 366 frames. Small blue circles are eye gaze points of operator.

(b) Sample frames from full synopsis video sequence of 185 frames.

(c) Sample frames from the synopsis of monitored parts. The result video contains 147 frames.



(d) Sample frames from the synopsis of where operator overlooks. It contains 130 frames.

Fig. 1. Sample frames from the first input video and its corresponding summaries.

compact video synopsis can be achieved after the clustering and overlapping actions but the learning stage takes very long time.

A fast approximation of non-linear video summarization method is proposed by Yildiz et al. (2008). The method works on 2D projections of the video volume instead of working on 3D volume itself. This projection reduces the complexity of the problem. Therefore, an efficient dynamic programming algorithm can be employed to optimize the video energy. The video energy is represented by an energy image after the projection. Every pixel of the energy image corresponds to a column of a video frame and a pixel has a higher energy value if there is an action on the corresponding column. A path with the minimum energy can be found and discarded on this image to decrease the video length. After removing these columns from the original video, the non-linearly summarized video is obtained.

The main contribution of the proposed system is the novel integration of the eye-gaze focus points with the improved real-time non-linear video summarization method of (Yildiz et al., 2008). We use a new efficient background subtraction algorithm that provides information about the number of frames to be discarded without limiting the summarization capacity. The overall system can be used with practical surveillance systems without complicating the task of the operator (Fig. 1). The system runs at real-time speeds on average hardware, which means that while the operator is working, the summary video of the overlooked (Fig. 1. (d)) or the attentively monitored (Fig. 1. (c)) video sections are already available at the end of the monitoring process. We also describe some improvements over our previous work (Vural & Akgul, 2009). Today's surveillance cameras generally produce low resolution videos that could not be useful for human identification (Keval & Sasse, 2006). So, the recorded videos are not accepted as an evidence in court(Sasse, 2010). Technological foresights on digital surveillance video systems indicate that most of the surveillance video cameras are going to be replaced with higher resolution ones. Our method reaches the real-time rates on high resolution videos by using parallelism and a better optimization method.

2. Background information

Video summarization methods are useful in video surveillance systems for decreasing the operational costs. They decrease the demand of manpower on video searching tasks as well as cutting down the storage costs. We use video summarization in surveillance somehow differently from the previous methods. We utilize the operator eye-gaze positions in summarizing the interesting sections of the surveillance videos, where interesting sections might include the overlooked or most attentively monitored sections. We employ a non-linear video summarization method for its efficiency and nonlinear treatment of its time dimension. The method depends on an observation of motion in real life activities. It assumes that almost all dynamic objects in surveillance scenes move horizontally on the ground and cameras are placed such that x axis of the camera reference frame is parallel to the ground. If we project the video volume onto the plane orthogonal to its y axis, the resulting projection reduces the size of the problem in exchange for losing the information of motion on the y axis (Fig. 2 Step-1). The projection keeps horizontal motion information on a 2D projection matrix, P. Despite the 3D nature of the video summarization problem, the method works on 2D projection matrix. The projection matrix P contains W x H elements for a video sequence of T frames each of which is W x T. Each element of matrix P represents a column of input video V, and their values are equal to the sum of the gray level pixels in the corresponding columns.

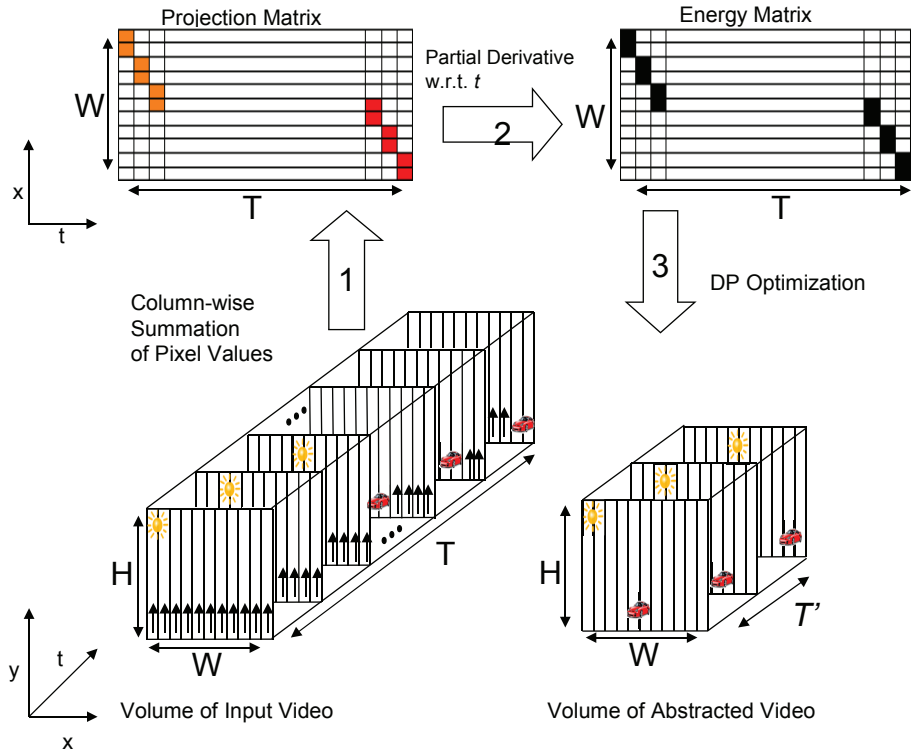


Fig. 2. Non-linear video summarization of an input video sequence with T frames. All frames of input video have width of W and height of H. A summarized video with T number of frames is obtained after 3 steps: 1- Projection of the columns, 2- Computation of Energy Matrix, 3- Optimization using dynamic programming.

$$P(w,t) = \sum_{h=1}^{H} V(w,h,t), \forall\ w,\ t,\ c,\ s.t.\ w \in [1,W],\ t \in [1,T]. \tag{1}$$

Although the projection operation reduces the problem size, the values in P are the summations of the pixel intensities and cannot be used alone in the optimizations. The second step of the summarization method constructs an energy matrix E with the same size of P(Fig. 2 Step-2). The elements of E are computed as a partial derivative of P with respect to time (Eq. 2) so the motion information is obtained from the brightness changes of the video columns.

$$E(w,t) = |\frac{\partial P(w,t)}{\partial t}|, \forall\ w,\ t,\ s.t.\ w \in [1,\ W],\ t \in [2,\ T]. \tag{2}$$

We briefly explain the dynamic programming based optimization here and leave the details to the next subsection (Fig. 2 Step-3). The method discards the video columns by running dynamic programming on the energy matrix E. While higher energy values in E mean there can be an action, lower energy values most probably represent background columns. The method uses dynamic programming to find a path with the minimum energy on E and

removes the corresponding pixels from the original video. These removed pixels make a surface in the 3D video volume which means that removing this surface makes the video shorter. Since the removed surface contains only the low energy pixels, background columns in the video are discarded. Matrix E is partially changed after the removal of the columns. New surfaces can be discarded by applying dynamic programming after computing the changed parts of matrix E. Applying these steps several times makes the video shorter and the video summary is obtained. Although the above method is similar to the non-linear image resizing method of Avidan & Shamir (2007), our employment of this method is original because we use it not for the image resizing but for video summarization. The energy matrix E_{img} of an image can be defined as the gradient magnitude of the original image I. Edges and textured regions in the image are most likely preserved.

Our video summarization method is based on the non-linear image resizing method of Avidan & Shamir (2007). In non-linear image resizing, an energy matrix of the original image is used for optimization. The energy matrix is specially formed from the original image such that matrix elements have higher energy values if they correspond to pixels with higher conceptual information. Dynamic programming runs on the energy matrix and finds a minimum energy path on the image. The pixels along the path are removed for shrinking the image. For horizontal shrinking, non-linear image resizing finds a vertical minimum energy path on the energy image and removes the pixels belonging to that path. Similarly, a horizontal minimum energy path is searched and its pixels are removed for vertical shrinking. Size of the new image is inversely proportional with the number of dynamic programming paths so for getting original image smaller more minimum energy paths must be found.

$$E_{img} = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \tag{3}$$

A vertical path on E_{img} should be found for horizontal shrinking and the path should have only one element for each row of the image. This rule enforces all rows to have the same number of pixels after every path removal. On a WxH image, a vertical path is defined as

$$S^v = \{col(h), h)\},\ s.t.\ \forall h, h \in [1, H],\ |col(h) - col(h-1)| \leq 1, \tag{4}$$

where $col(h)$ is the column position of path element on row h. A vertical path S^v is composed of h points and the neighboring points of the path can have at most 1 displacement in the horizontal direction. Similarly, a horizontal path S^h is defined as

$$S^h = \{(w, row(w))\},\ s.t.\ \forall w, w \in [1, W],\ |row(w) - row(w-1)| \leq 1. \tag{5}$$

Finding the vertical or the horizontal minimum energy paths on E_{img} and removing the corresponding pixels will shrink the image in the desired dimension. The minimum energy path is found using dynamic programming. Dynamic programming first fills a table M with the cumulative cost values of the paths then back traces on this table to find the actual path elements. The values of M are computed using the following recursion

$$M(w,h) = E_{img}(w,h) + min\{M(w-1,h-1), M(w,h-1), M(w+1,h-1)\}. \tag{6}$$

When M is fully constructed, the minimum costs for the paths are placed at the last row of M. The minimum cost value of the last row equals to the total cost of the minimum energy vertical path and the position of the minimum cost value gives the last element of the path.

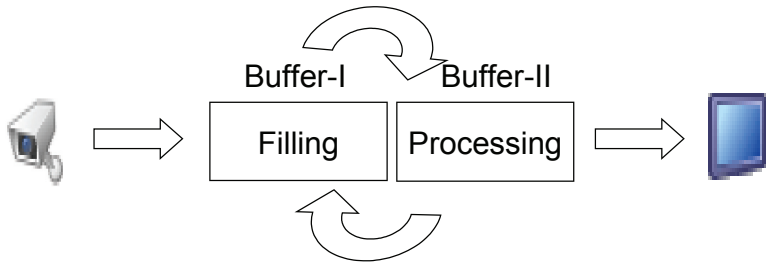


Fig. 3. Our method works on two buffers for handling real-time video summarization.

Dynamic programming finds all path elements by back tracing from that position. At the end of this process we have the minimum path across the energy image. All pixels belonging to this path are discarded to shrink the image by one column.
This method can be used in 3D space-time video volume as well as 2D images. A non-linear video summarization from the space-time video volume can be achieved by shrinking the time dimension. A naive approach would search a 3D surface of pixels with least motion information instead of a 2D path. The video summary can then be produced by discarding a surface with the minimum energy but finding such a surface with dynamic programming would take exponential time. This problem is solved by projecting the video volume onto a plane orthogonal to its x or y axes (Yildiz et al., 2008).

3. The method

Our method employs the projection technique used in (Yildiz et al., 2008) to obtain a projection matrix. We then use a novel frequency based background subtraction method on the projection matrix. The video sections with motion information in the background matrix B are then filtered according to the eye-gaze positions obtained from the operator. The filtering can be performed to produce overlooked sections or the sections that have the operator focus. At the last step, we run the dynamic programming algorithm for producing the video summary.
We use two buffers for the real-time processing of the video. Each buffer is processed by a separate process. One of the processes fills its buffer with video frames and computes the corresponding row of projection matrix P just after grabbing the frame. Since computing projection of a frame does not depend on other frames, one process can handle grabbing and projection together. Once the first process fills its buffer, it hands the current buffer over to the second process and it starts filling the other. The second process begins processing the full buffer by computing energy matrix from the present projection matrix and continuously finds the minimum energy paths for summarizing the video.
The following subsections include novel contributions of our method for the background subtraction and the employment of eye-gaze positions.

3.1 Frequency based background subtraction

Although the video abstraction method of (Yildiz et al., 2008) is fast, direct employment of this method in our application has several problems. First, computed values on E are the absolute differences of total intensity values between two consecutive columns (Eq. 2). The value

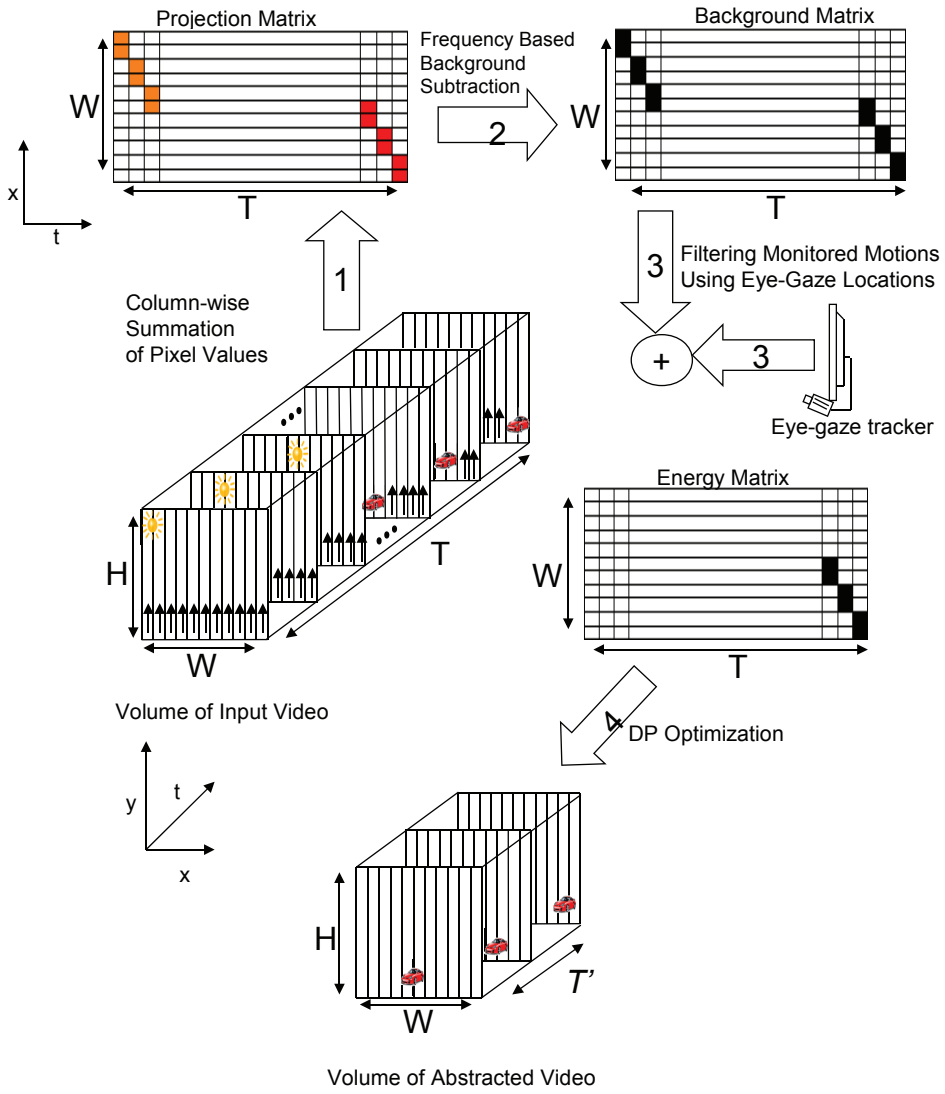


Fig. 4. Non-linear video summarization of the interesting sections contains 4 steps: 1- Projection of the video columns, 2- Background subtraction, 3- Computing Energy Matrix E_{gaze} considering eye-gaze positions, 4- Optimization using dynamic programming.

gets larger while the intensity dissimilarity between the moving object and the background increases. Second, the system produces positive costs for the video columns with no motion information, if there are lighting variations. Third, the system cannot determine how many frames have to be discarded because it does not know what values represent an action. Finally, the system does not have any mechanisms of filtering according to eye-gaze positions.
Our new frequency based background subtraction method produces a binary map B of background and actions. Values of the projection matrix elements are scaled to the interval of $[0, S]$ for a scaling parameter S. This scaling operation limits the maximum value with a relatively small number and lets us use a histogram based fast frequency transform.
Our method counts the number of scaled intensity values for the rows of matrix P using an histogram array A with a size of S. The values of histogram array for a row $w \in [1, W]$ are computed as follows:

$$A_w[P(w,t)] = A_w[P(w,t)] + 1 \;\; \forall t \; s.t., \; 1 \leq t \leq T. \tag{7}$$

The last step of computing frequency based background matrix is extracting the background and actions. We use a technique similar to one described by Zhang & Nayar (2006), for extracting background from the video frames. Since the histogram values for the action pixels of matrix P is expected to be less than the pixels of the background, a simple thresholding method can be used to form the background matrix B.

$$B(w,k) = \begin{cases} ACTION & if \; A_w(k) \leq threshold_1, \\ BACKGROUND & otherwise. \end{cases} \tag{8}$$

3.2 Tracking eye-gaze positions of human operator

The proposed system requires both background matrix B and eye-gaze positions of the operator for computing energy matrix of interesting video sections (Fig. 4 Step-3). Although we use the eye-gaze tracker of LCTechnologies(LCTechnologies (1997)), any eye-gaze tracker (Hutchinson et al., 1989; Morimoto & Mimica, 2005) that does not disturb operators would work with our system. The tracker communicates with our application and returns the x and y positions of the operator's eye-gaze position for each video frame. First, we label each frame as 'monitored' or 'not monitored' by checking if the eye-gaze position of the operator is within the display area.

$$L(t) = \begin{cases} monitored & if(G_x(t) \in [0, W] \wedge G_y(t) \in [0, H], \\ not \; monitored & otherwise, \end{cases} \tag{9}$$

where $L(t)$ is the label of the frame, $G_x(t)$ and $G_y(t)$ are x and y positions of eye-gaze position at time t. We preprocess $G_x(t)$ and $G_y(t)$ before they are used in Eq. 9 for suppressing the effect of eye blinking. Our system uses an outlier detection approach for determining the frames with eye blinking. For such blinking frames the last valid eye-gaze position is applied as G_x and G_y.
The above formulations are sufficient to find if the operator misses the whole frame. For such cases, our dynamic programming based abstraction method includes the action sections of the frame in the video summary because it is known that none of the actions are monitored by the operator.

If the eye-gaze positions of the operator is on the display area, we need a mechanism of what sections of the video the operator is focused on. Although sensing and tracking actions generally can be done fast, operators cannot focus to see all the actions on a monitor if there are several independently moving objects (Sears & Pylyshyn, 2000). Detecting such a situation is also important to understand if the action is seen by operator or not. Human visual system has a good and efficient mechanism for tracking moving objects. The eye focuses near the moving object if there is only one object (Fig. 8. (a)). It focuses at the center of moving objects if there are more than one related object (Fehd & Seiffert, 2008) (Fig. 8. (b)). We also observe this behavior in our experiments, which led us to use a circular attention window for covering action sections. A circular area around the eye gaze position is assumed as the visual field where a human can catch actions. The radius of the circle is determined experimentally in our work and we set it as quarter of the screen dimensions.

Our summarization method uses a weight array ω for ignoring or accepting the video sections according to eye-gaze positions of the operator. The weight array with values larger than 1 increases the acceptance chance (ω^+) of the section and the values smaller than 1 decreases the chance (ω^-). These arrays are filled with constant numbers, however our formulations do not prevent any employment of varying numbers that increases the weights of the center pixels. Since our system cannot discard a video column partially due to the projection of the 3D video volume to a 2D projection image, vertical weighting is unnecessary. Therefore, using a simple weight array is sufficient. The ω contains $2r + 1$ elements where r is the radius of attention circle. The system can have either one of two different special abstracts using one of the weight arrays above. The abstract video can show either 'attentively monitored' or 'overlooked' parts depending on which weight array is used.

We construct our eye-gaze based energy matrix E_{gaze} from background matrix B using a weight array ω.

$$E_{gaze}(w,t) = \begin{cases} B(w,t)\ \omega[G_x(t) - w] & if\ |G_x(t) - w| \leq r, \\ B(w,t) & otherwise. \end{cases} \tag{10}$$

The new energy matrix E_{gaze} is the matrix that will be used to run the dynamic programming based video summary method.

3.3 A more efficient and parallelizable method

It is expected that most of the surveillance cameras are going to be replaced with higher resolution cameras(Sasse, 2010). New generation cameras produce high quality videos which are more useful for accurate recognition of objects and actions but processing these videos will require more CPU power. Some of the existing surveillance video synopsis methods work in delayed real-time with today's surveillance cameras but further improvements are required for handling higher resolution videos. Analysis of our existing synopsis method shows that the method is highly parallelizable and dynamic programming based optimization can be replaced with an efficient binary optimization method. In order to speed up the synopsis system, we use binary energy images more effectively both for finding minimum energy paths and for the regeneration of synopsis video.

The slowest part of the existing method is the regeneration of synopsis video after removing the minimum energy paths. Video columns that do not correspond to the minimum energy path elements are moved on time axis to reconstruct synopsis video. Moving the input video columns to their new locations in the synopsis video volume is a time consuming

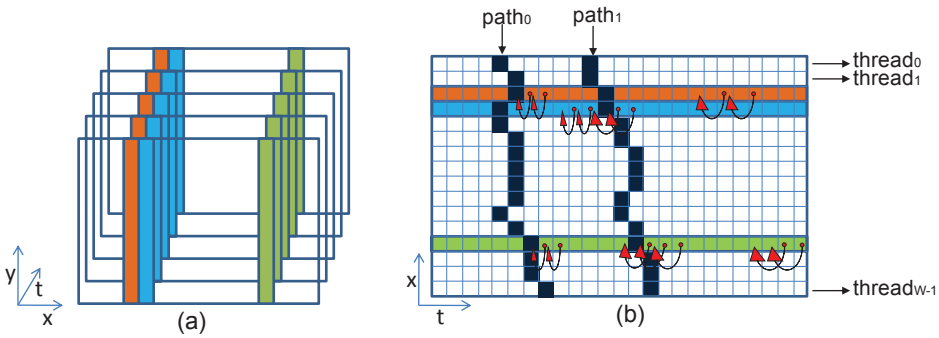


Fig. 5. (a) An input video volume. Each different color shows columns with different column indexes. (b) Energy matrix E of (a). Two sample paths are found so the video will be two frames shorter. One of these paths is used for constituting the background frame $F_{background}$. Each column index is represented with a different color on the rows of the energy matrix E. One thread is assigned for each row of E. Sample movements of video columns are also shown.

task. Existing non-linear synopsis methods spend most of their computation times on the reconstruction phase. In experiments, it is shown that total processing time decreases when the number of extracted minimum energy paths increases. The methods get faster when the number of moving columns is smaller. While the reconstruction of synopsis video is the most time consuming phase, it is important to improve this phase for speeding up the whole synopsis system. We propose two improvements for this phase: first improvement is to prevent to movement of redundant columns and the second improvement is using parallel programming techniques for the reconstruction of synopsis video.

Although the system ensures that a path element represents a background column, the video columns to be moved do not only contain actions but also background columns which do not include any minimum energy path(Fig. 7). Fast non-linear synopsis method of Yildiz et al. uses a real valued energy matrix E_r and this map does not know anything about if a map element represents an action or background (Yildiz et al., 2008). Thus, this method has to move all columns for the reconstruction. A large number of video columns can be skipped in the moving process by using the binary energy matrix E_b effectively. First, a background image is obtained using one of the minimum energy paths (Fig. 5(b)). Video columns that correspond to a zero energy path constitute a background frame $F_{background}$ by using Eq. 11.

$$F_{background}(i,j) = V(i,j,path[i].t), \forall\ i,\ j,\ s.t.\ i \in [1,W],\ j \in [1,H] \tag{11}$$

where path[i].t is the frame number of the i^{th} element of the minimum energy path stack.

We first create a synopsis video volume whose frames are $F_{background}$ and then only move the action columns to their new locations in the synopsis video volume. The moving operation is similar to the moving operation of (Vural & Akgul, 2009).

The second improvement on the synopsis video volume reconstruction is using parallelism. An input video column can only be moved to a synopsis video column on the same column index (Fig. 5(a)). In other words, a column can only be moved on the time axis. This operation is independent from other columns with different w indexes. We assign one thread for each

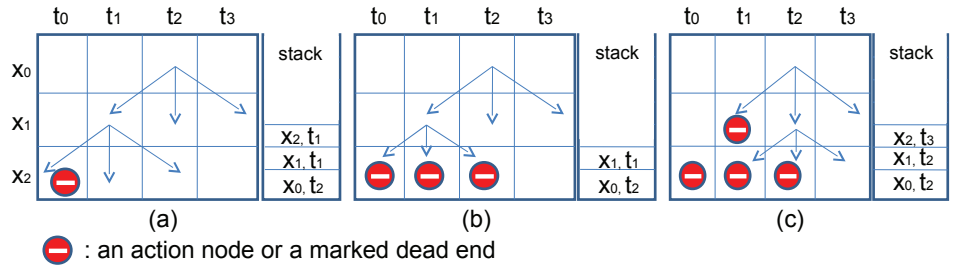


Fig. 6. (a) A path starting from (x_0,t_2) finishes on (x_2,t_1). A stack holds the path elements. (b) Dead-end: (x_1,t_1) is chosen as a next path element from the node (x_0,t_2) but there are no available path elements in the neighborhood of (x_1,t_1). (c) (x_1,t_1) is removed from the stack and marked as an unavailable node. (x_1,t_2) is chosen instead of (x_1,t_1). A complete path is found after this choice.

row of the energy matrix E(Fig. 5(b)). Each thread handles the movement of all columns in a row of binary energy matrix E.

Another time consuming part of the existing non-linear synopsis methods is the DP optimization. In DP optimization it is required to find total costs of all minimum energy paths. When a path is found and removed from the energy matrix E, neighborhood of video columns is changed and recalculations of total path energies are needed for a partial region of E. DP finds all the paths and then it requires a back-tracing for the one with the minimum energy to find path elements. A more efficient method that uses binary energy matrix E can be applied instead of DP. Total energy of a permitted path on a binary energy map must be zero. While we know the total energy, computation of path's total energy is redundant. A path can only contain elements which represent the background columns with zero energy values. The proposed method tries to find paths for all background elements of the first row of the binary energy matrix E. A stack data structure is used for storing path elements. The first element of the stack is chosen from the first row of the binary energy matrix E. The method seeks a path for each background element of the first row. The stack is initialized for each path and the next background element from the first row of E is added to the stack. The top element of the stack is the active node. From the active node, next element of the path can be one of the three neighbors on the next row. A background element from the active node's neighborhood is added to the stack. A path is completed when the stack includes an element from the last row of the energy image E (Fig. 6(a)). In some cases, there can be no available background nodes in the neighborhood of an active node as seen in Fig. 6(b). If there is not any path from an active node, the method marks that node as a dead-end node. A dead-end node is removed from the stack and a new path is tried from the previous active node (Fig. 6(c)). A marked dead-end node loses its availability and further paths never traverse it again.

Although there are some other parallelism mechanisms could be found for the computation of the projection matrix P and the energy matrix E, their effect will be marginal on the total processing time. A pipeline mechanism could also be used while computing the projection matrix P. Each row of the projection matrix P is only depended on one input video frame, so a row of P could be computed just after the frame is graped. We tested the proposed

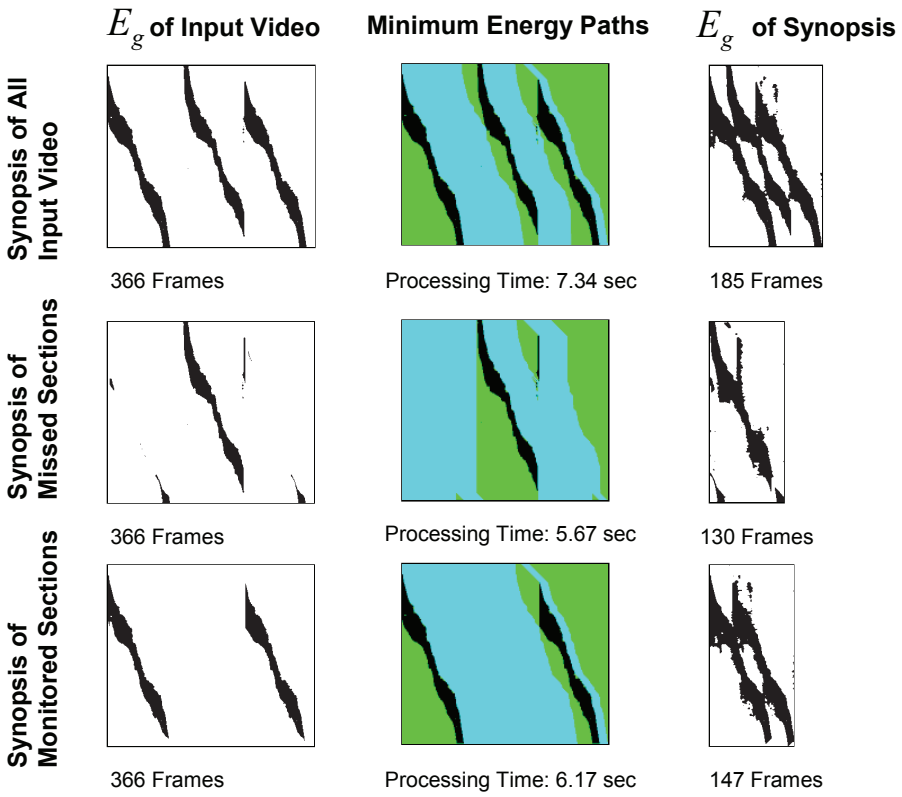


Fig. 7. E_{gaze} images and the minimum energy paths of the first input video. Black regions show action trajectories and the cyan colored regions represent minimum energy paths. Green regions are also representing background columns like cyan regions but the difference is that there is not any minimum energy path passing from green elements. One improvement on synopsis video reconstruction phase skips over working on these columns. Processing times are for single threaded method running on 3.2Ghz CPU.

improvements on different computers with varied sized sample videos. The results show that our method reaches the real-time rates for high resolution images by using above improvements.

4. Experiments

We group our experiments into two parts. The videos of these experiments can be viewed at http://vision. gyte. edu. tr/projects. php?id=5.In the first group we analyze how humans track and sense moving objects. This analysis is important to understand the relationship between eye movements and observed actions. We prepared six synthetic movies with different number of moving objects and motion characteristics to test on a group of people. The experiments show us how an eye-gaze position gets its initial position when an action

	Intel P4 3.2Ghz with hyper-threading		Intel QuadCore 2.5Ghz	
	Single Threaded	Multi Threaded	Single Threaded	Multi Threaded
320x240	7.349	2.532	4.578	0.859
640x480	23.51	10.17	16.06	2.844
1024x768	104.3	39.98	48.30	10.81
1920x1080	182.1	87.76	126.4	30.07

Table 1. Running times (sec) of single thread method of (Vural & Akgul, 2009) and improved multi-threaded method on different resolutions. Experiments are done on two different computer setups.

appears and how the tracking is continued. The eye moves totteringly when it first recognizes a moving object and nearly after two seconds all the subjects' eyes find a stable trajectory for tracking. Tracking is more complex for multiple moving objects on different sections of the monitor. Although most of the subjects prefer to track as many objects as possible, eyes move towards crowded sections of the monitor (Fig. 8. (c)). This initial latency and tottering can cause overlooking some actions. We also observed that our experiments support the thesis about multiple moving objects in (Fehd & Seiffert, 2008). Human eyes are rather focused around moving objects instead of focusing directly on the objects conference (Fig. 8. (a,b)). Therefore using an attention area to represent this adjacency is required and we represent this area in a circular form.

In the second group of experiments, we tested our method on two different scenarios of surveillance videos with varied resolutions. We show the results of our video summarizations and compare them with each other according to their frame numbers and processing times. The videos are recorded in our laboratory and we instruct our operators to monitor some actions and overlook others. Our videos are at 15fps and the resolutions are 320 x 240, 640x480, 1024x768 and 1920x1080.We select the scaling parameter S as 255 and $threshold_1$ of Eq. 8 as 5 for all our experiments. We tested our methods on two different computer setups. First computer is a hyper-threaded Intel Pentium-4 3.2GHz PC with 1GB of memory and the second one has 2GB memory and an Intel QuadCore 2.5Ghz CPU.

In the first video a person walks and another person traces nearly the same route after the first person leaves the field of view of the camera. The first person then again walks in the room. We instructed our operator to direct his eye-gaze out of the display area when the second person appears on the screen. Sample frames from this scenario are shown in Fig. 1. We also show images of the minimum energy paths and the E_{gaze} matrices of both input and result videos (Fig. 7). First input video is 24 second long and our single threaded method summarizes the overlooked sections of it in 5.67 seconds with the DP optimization. The processing time of attentively monitored sections is a little longer than the overlooked parts and it takes 6.17 seconds. The processing time of the video decreases when the number of minimum energy paths increases for the single threaded method of (Vural & Akgul, 2009). This shows that the most time consuming part of the system is reconstruction of the video volume for summarization. Time requirement of this step increases with the number of frames in the video.

We compare the running time of single-threaded synopsis method of Vural et. al (2009) with the proposed multi-threaded method. In Table. 1 we tested several different resolutions of first input video on two different computers. We run each method ten times and the values

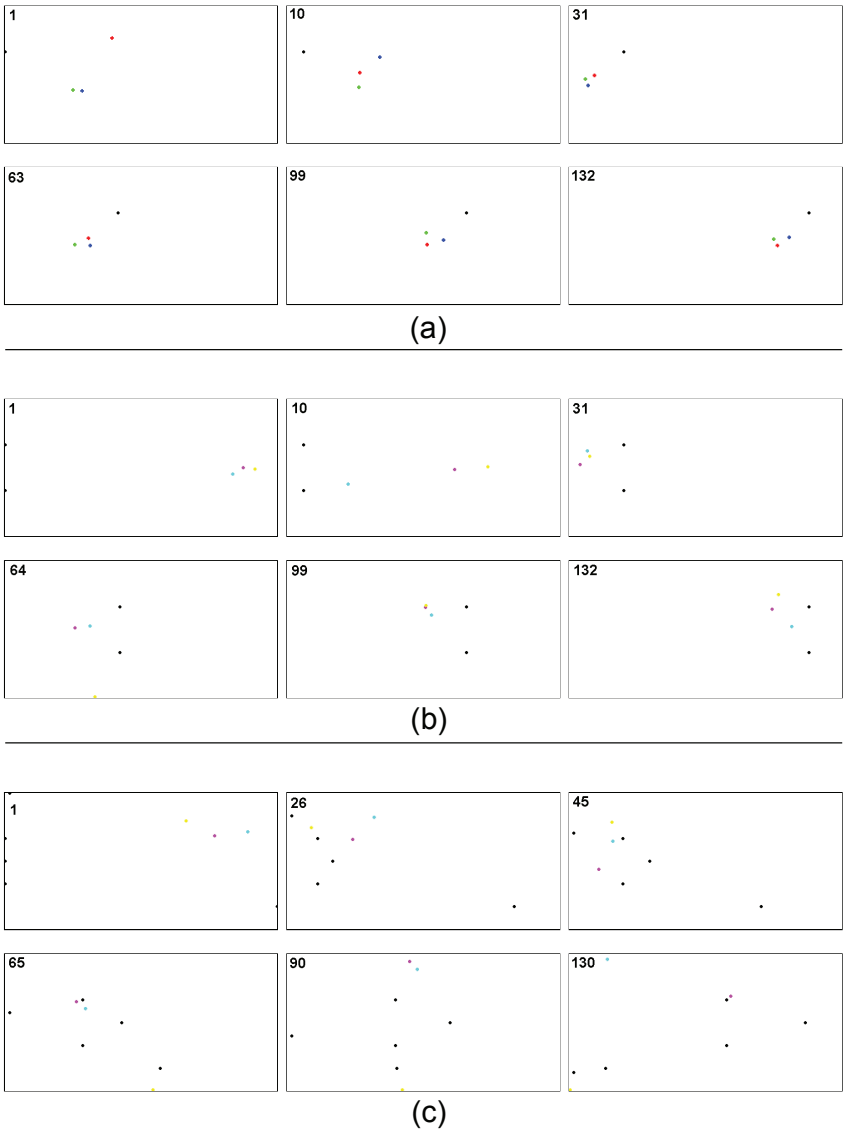


Fig. 8. How an eye tracks moving objects: Black circles are moving objects and the circles with other colors represent the eye gaze points of different subjects. (a) Tracking one moving object, (b) Tracking two objects moving same direction. (c) Tracking five objects moving different directions.

	A	B	C	D
Full synopsis (1920x1080)	126.4	101.0 (125%)	52.59 (240%)	30.07 (420%)
Full synopsis (1024x768)	48.30	39.94 (121%)	22.38 (216%)	10.81 (447%)
Full synopsis (320x240)	4.578	2.375 (193%)	1.094 (418%)	0.859 (533%)
Overlooked (320x240)	3.962	1.782 (222%)	1.031 (384%)	0.750 (528%)
Monitored (320x240)	4.047	2.031 (199%)	1.203 (336%)	0.843 (480%)

Table 2. Speeding up effects of the improvements on the first input video. Column A: Single-threaded synopsis method, Column B: Single-threaded method without DP optimization, Column C: Multi-threaded method without DP optimization, Column D: Multi-threaded method without DP optimization and with skipping redundant video columns from moving. Running times are (sec) achieved on Intel QuadCore 2.5 CPU with 2GB memory. Percentages represent the speeding up ratio of that column from the Column A.

in the table are averages. The experiments show that the multi-threaded synopsis method is at least the number of CPU cores times faster than the single threaded method. While the limited memory sizes limit the speeding up, the improvement is nearly two times more on lower resolution videos.

We also show how each improvement effects on the running times. In Table. 2. running times of single threaded synopsis method and the effects of different combination of improvements are shown. In higher resolution videos dynamic programming based optimization takes less percentage of time on total running time. So, the removing DP can only effect up to 240% over single-threaded method. Multi-threading decreases running time between 50% to 75%. The last column of the table shows multi-threaded synopsis method without DP and skipping background columns from moving. The improvement is up to 533% and this improvement is proportional with the number of background nodes which are not marked as minimum energy path elements. All of our running times on the last column except for 1920*1080 resolution are shorter then the original input video length. Our multi-threaded method runs on delayed real-time for other resolutions of video on 15 fps. The method reaches only 11 fps on 1920x1080 resolution but this speed is also acceptable on today's high resolution surveillance cameras. High resolution surveillance cameras works at 25 fps for 1024x768 (768.) resolution and at 10fps for 1920x1080 (1080p). In the last table (Table. 3 we show the running time results of second input video.

Our last experiment is for analyzing the behavior of our system when an operator overlooks an action while watching another action on the same monitor (Fig. 9.) In this scenario a bag is stolen but our operator watches the other side of the monitor. We then show the rubbery again to the operator by processing the 24 second long input video in only 4.39 seconds. There are some artifacts in summarized videos. These artifacts occur because of the constant radius of visual attention circle. If the attention circle covers only some part of the action, the other parts can be discarded. One solution to this problem could be a simple motion segmentation module that prevents segments from partial omission. We prefer not to use such a mechanism due to the real-time requirements of our system.

	A	B	C	D
Full synopsis (320x240)	3.782	2.406 (157%)	1.297 (292%)	0.797 (475%)
Overlooked (320x240)	3.632	1.516 (240%)	0.969 (375%)	0.698 (520%)
Monitored (320x240)	3.657	2.324 (157%)	1.188 (308%)	0.765 (478%)

Table 3. Speeding up effects of the improvements on the second input video. Column A: Single-threaded synopsis method, Column B: Single-threaded method without DP optimization, Column C: Multi-threaded method without DP optimization, Column D: Multi-threaded method without DP optimization and with skipping redundant video columns from moving. Running times are (sec) achieved on Intel QuadCore 2.5 CPU with 2GB memory. Percentages represent the speeding up ratio of that column from the Column A.

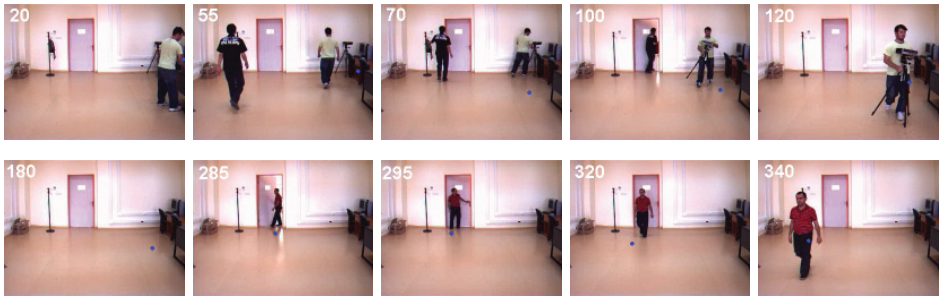
5. Conclusions

We introduced a novel system for the real-time summarization of the high resolution surveillance videos under the supervision of an surveillance operator. The system employs an eye-gaze tracker that returns the focus points of the surveillance operator. The resulting video summary is an integration of the actions observed in the surveillance video and the video sections where the operator pays most attention or overlooks. The unique combination of the eye-gaze positions with the non-linear video summaries results in a number of important advantages: First, it is possible to review what actions happened in the surveillance video in a very short amount of time. If there are many operators monitoring different cameras, the supervisor of the surveillance system can check what the operators observed without going through all the videos. Second, it is possible to review the overlooked actions of the surveillance videos efficiently. Finally, as a side benefit of the second advantage, it is possible to evaluate the performance of the surveillance operators by analyzing the overlooked sections of the videos. This advantage makes it possible to adjust the number of operators, their work durations and the work environment conditions.

The proposed system requires the tracking of the operators gaze for the gaze positions, which might seem like a disturbance for the operator. However, eye-gaze tracking is becoming very popular and seamless systems started to appear in the market for very low costs. We expect that the advantages of the proposed system far exceed the disadvantage of the added eye-gaze tracker.

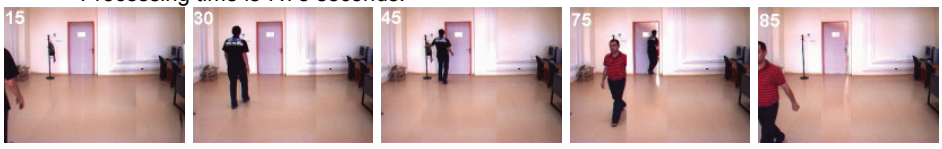
Another limitation of the system might be the employment of the 3D video projection to the 2D images that loses some of the action information. However, our experiments with the real surveillance scenes indicated that this is not a serious problem because in surveillance videos most of the action happens on a horizontal plane and vertical actions are always coupled with horizontal actions. The experiments we performed on real and synthetic videos indicated that our system is actually works in the real world and can easily be employed in practice.

Although the system is formulated and the experiments are performed under the assumption that only the video sections with movements are interesting, the system can be easily modified to change what is interesting. There are systems that classify the video sequences as interesting or not interesting, which could be easily integrated with our system for other types of video summaries.



(a) Sample frames from an input video sequence of 359 frames. Small blue circles are eye gaze points of operator.

(b) Sample frames from full synopsis video sequence of 209 frames. Processing time is 7.78 seconds.



(c) Sample frames from the synopsis of where operator overlooks. It contains 95 frames. The summary is extracted in 4.39 seconds.

(d) Sample frames from synopsis of monitored parts. The result video contains 137 frames and is processed in 5.47 seconds.

Fig. 9. Sample frames from the second input video and its corresponding abstracted videos. Processing times are for single threaded method on 3.2Ghz CPU.

6. Acknowledgements

This work is supported by TUBITAK Project 110E033.

7.References

Acha, A. R., Pritch, Y. & Peleg, S. (2006). Making a long video short: Dynamic video synopsis, *IEEE Computer Vision and Pattern Recognition or CVPR*, pp. I: 435–441.

Ahmad, I., He, Z., Liao, M., Pereira, F. & Sun, M. (2007). Special issue on video surveillance, *Circuits and Systems for Video Technology, IEEE Transactions on* 17(9): 1271–1271.

Avidan, S. & Shamir, A. (2007). Seam carving for content-aware image resizing, *ACM Trans. Graph.* 26(3): 10.

Buono, P. & Simeone, A. L. (2010). Video abstraction and detection of anomalies by tracking movements, *AVI '10: Proceedings of the International Conference on Advanced Visual Interfaces*, ACM, New York, NY, USA, pp. 249–252.

Chen, B. & Sen, P. (2008). Video carving, *In Short Papers Proceedings of Eurographics*.

Choudhary, V. & Tiwari, A. K. (2008). Surveillance video synopsis, *ICVGIP '08: Proceedings of the 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, IEEE Computer Society, Washington, DC, USA, pp. 207–212.

D., S. G. J. (2004). Behind the screens: Examining constructions of deviance and informal practices among cctv control room operators in the uk., *Surveillance and Society*, Vol. 2(2/3), pp. 376–395.

Dick, A. R. & Brooks, M. J. (2003). Issues in automated visual surveillance, *In International Conference on Digital Image Computing: Techniques and Applications*, pp. 195–204.

Fehd, H. M. & Seiffert, A. E. (2008). Eye movements during multiple object tracking: Where do participants look, *Cognition* 108(1): 201–209.

Franconeri, S. L., Alvarez, G. A. & Enns, J. T. (2007). How many locations can be selected at once?, *J Exp Psychol Hum Percept Perform* 33(5): 1003–1012.

Green, M. (1999). The appropriate and effective use of security technologies in us schools. a guide for schools and law enforcement, *Technical report*.

Gryn, J. M., Wildes, R. P. & Tsotsos, J. K. (2009). Detecting motion patterns via direction maps with application to surveillance, *Computer Vision and Image Understanding* 113(2): 291 – 307.

Hampapur, A., Brown, L., Feris, R., Senior, A., Shu, C., Tian, Y., Zhai, Y. & Lu, M. (2007). Searching surveillance video, *AVSBS07*, pp. 75–80.

Hu, W., Tan, T., Wang, L. & Maybank, S. (2004). A survey on visual surveillance of object motion and behaviors, *IEEE Trans. Syst., Man, Cybern* pp. 334–352.

Hutchinson, T. E., White, K. P., Martin, W. N., Reichert, K. C. & Frey, L. A. (1989). Human-computer interaction using eye-gaze input, *Systems, Man and Cybernetics, IEEE Transactions on* 19(6): 1527–1534.

Iannizzotto, G., Costanzo, C., Rosa, F. L. & Lanzafame, P. (2005). A multimodal perceptual user interface for video-surveillance environments, *ICMI*, pp. 45–52.

Jacob, R. (1991). The use of eye movements in human-computer interaction techniques: What you look at is what you get, *ACM Transactions on Information Systems* 9(3): 152–169.

Jaimes, R., Pelz, J., Grabowski, T., Babcock, J. & fu Chang, S. (2001). Using human observers' eye movements in automatic image classifiers, *Proceedings of SPIE Human Vision and Electronic Imaging VI*.

Jing, Z. & Lansun, S. (2008). A personalized image retrieval based on visual perception, *Journal of Electronics (China)*, Vol. 25.

Keval, H. & Sasse, M. A. (2006). Man or gorilla? performance issues with cctv technology in security control rooms, *16th World Congress on Ergonomics Conference, International Ergonomics Association* .

Kim, C. & Hwang, J.-N. (2000). An integrated scheme for object-based video abstraction, *MULTIMEDIA '00: Proceedings of the eighth ACM international conference on Multimedia*, ACM, New York, NY, USA, pp. 303–311.

Komlodi, A. & Marchionini, G. (1998). Key frame preview techniques for video browsing, *DL '98: Proceedings of the third ACM conference on Digital libraries*, ACM, New York, NY, USA, pp. 118–125.

Koskela, H. (2000). The gaze without eyes: Video-surveillance and the nature of urban space, *Progress in Human Geography* 24(2): 243–265.

LCTechnologies (1997). The eyegaze communication system.
URL: *http://www.eyegaze.com*

Li, F. C., Gupta, A., Sanocki, E., wei He, L. & Rui, Y. (2000). Browsing digital video, *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, New York, NY, USA, pp. 169–176.

Li, T., Mei, T., Kweon, I.-S. & Hua, X.-S. (2009). Multi-video synopsis for video representation, *Signal Process.* 89(12): 2354–2366.

Li, Y., Li, Y., Zhang, T., Zhang, T., Tretter, D. & Tretter, D. (2001). An overview of video abstraction techniques, *Technical report*, Imaging Systems Laboratory, HP Laboraties, Palo Alto.

López, M. T., Fernández-Caballero, A., Fernández, M. A., Mira, J. & Delgado, A. E. (2006). Visual surveillance by dynamic visual attention method, *Pattern Recogn.* 39(11): 2194–2211.

Morimoto, C. H. & Mimica, M. R. M. (2005). Eye gaze tracking techniques for interactive applications, *Comput. Vis. Image Underst.* 98(1): 4–24.

Norris, C. & Armstrong, G. (1999). Cctv and the social structuring of surveillance, *Surveillance of Public Space: CCTV, Street Lighting and Crime Prevention.*, Monsey: Criminal Justice Press.

Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S. & Carey, T. (1994). *Human-Computer Interaction*, Addison-Wesley Longman Ltd., Essex, UK, UK.

Pritch, Y., Ratovitch, S., Hendel, A. & Peleg, S. (2009). Clustered synopsis of surveillance video, *AVSS '09: Proceedings of the 2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, IEEE Computer Society, Washington, DC, USA, pp. 195–200.

Pylyshyn, Z. W. & Storm, R. W. (1988). Tracking multiple independent targets: evidence for a parallel tracking mechanism., *Spatial vision* 3(3): 179–197.

R., A., P., M., D., P. & B., M. A. (2004). Attention and expertise in multiple target tracking, *Applied Cognitive Psychology* 18: 337–347.

Sasse, M. A. (2010). Not seeing the crime for the cameras?, *Commun. ACM* 53(2): 22–25.

Sears, C. R. & Pylyshyn, Z. W. (2000). Multiple object tracking and attentional processing, *Canadian Journal of Experimental Psychology* 54(1): 1–14.

Siebel, N. T. & Maybank, S. J. (2004). The advisor visual surveillance system, *in* M. Clabian, V. Smutny & G. Stanke (eds), *Proceedings of the ECCV 2004 workshop "Applications of Computer Vision" (ACV'04), Prague, Czech Republic*, pp. 103–111.

Slot, K., Truelsen, R. & Sporring, J. (2009). Content-aware video editing in the temporal domain, *SCIA '09: Proceedings of the 16th Scandinavian Conference on Image Analysis*, Springer-Verlag, Berlin, Heidelberg, pp. 490–499.

Snoek, C. G. M. & Worring, M. (2005). Multimodal video indexing: A review of the state-of-the-art, *Multimedia Tools Appl.* 25(1): 5–35.

Steiger, O., Cavallaro, A. & Ebrahimi, T. (2005). Real-Time Generation of Annotated Video for Surveillance, *Proceedings of IEEE Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS 2005*, ISCAS, SPIE.

Truong, B. T. & Venkatesh, S. (2007). Video abstraction: A systematic review and classification, *ACM Trans. Multimedia Comput. Commun. Appl.* 3(1): 3.

Vural, U. & Akgul, Y. S. (2009). Eye-gaze based real-time surveillance video synopsis, *Pattern Recogn. Lett.* 30(12): 1151–1159.

Yildiz, A., Ozgur, A. & Akgul, Y. (2008). Fast non-linear video synopsis, *23rd of the International Symposium on Computer and Information Sciences, Istanbul, Turkey* . URL: *http://vision.gyte.edu.tr/projects.php?id=5*

Zhang, L. & Nayar, S. (2006). Projection defocus analysis for scene capture and image display, *ACM Trans. Graph.* 25(3): 907–915.