Nonchronological Video Synopsis and Indexing

Yael Pritch, Alex Rav-Acha, and Shmuel Peleg, *Member*, *IEEE*

Abstract—The amount of captured video is growing with the increased numbers of video cameras, especially the increase of millions of surveillance cameras that operate 24 hours/day. Since video browsing and retrieval is time consuming, most captured video is never watched or examined. Video synopsis is an effective tool for browsing and indexing of such a video. It provides a short video representation, while preserving the essential activities of the original video. The activity in the video is condensed into a shorter period by simultaneously showing multiple activities, even when they originally occurred at different times. The synopsis video is also an index of the original video by pointing to the original time of each activity. Video synopsis can be applied to create a synopsis of endless video streams, as generated by webcams and by surveillance cameras. It can address queries like "Show in one minute the synopsis of this camera broadcast during the past day." This process includes two major phases: 1) an online conversion of the endless video stream into a database of objects and activities (rather than frames) and 2) a response phase, generating the video synopsis as a response to the user's query.

Index Terms—Video summary, video indexing, video surveillance.

1 INTRODUCTION

EVERYONE is familiar with the time-consuming activity involved in sorting through a collection of raw video. This task is time consuming since it is necessary to view the video in order to determine if anything of interest has been recorded. While this tedious task may be feasible in personal video collections, it is impossible when endless video, as recorded by surveillance cameras and webcams, is involved. It is reported, for example, that, in London alone, there are millions of surveillance cameras covering the city streets, each camera recording 24 hours/day. Most surveillance video is therefore never watched or examined. Video synopsis aims at taking a step toward sorting through video for summary and indexing and is especially beneficial for surveillance cameras and webcams.

The proposed video synopsis is a temporally compact representation of video that enables video browsing and retrieval. This approach reduces the spatiotemporal redundancy in video. As an example, consider the schematic video clip represented as a space-time volume in Fig. 1. The video begins with a person walking on the ground and, after a period of inactivity, a bird is flying in the sky. The inactive frames are omitted in most video abstraction methods. Video synopsis is substantially more compact, playing the person and the bird simultaneously. This makes optimal use of image regions by shifting events from their original time intervals to other time intervals when no other activities take place at these spatial locations. Such manipulations relax the chronological consistency of events, an approach also used in [27].

The basic temporal operations in the proposed video synopsis are described in Fig. 2. Objects of interest are defined and are viewed as tubes in the space-time volume. A temporal shift is applied to each object, creating a shorter video synopsis while avoiding collisions between objects and enabling seamless stitching.

The video synopsis suggested in this paper is different from previous video abstraction approaches (reviewed in Section 1.1) in the following two properties: 1) The video synopsis is itself a video, expressing the dynamics of the scene, and 2) to reduce as much spatiotemporal redundancy as possible, the relative timing between activities may change. The latter property is the main contribution of our method.

Video synopsis can make surveillance cameras and webcams more useful by giving the viewer the ability to view summaries of the endless video in addition to the live video stream. To enable this, a synopsis server can analyze the live video feed for interesting events and record an object-based description of the video. This description lists, for each webcam, the interesting objects, their duration, their location, and their appearance. In a 3D space-time description of the video, each object is represented by a "tube."

A query that could be answered by the system may be similar to "I would like to watch in one minute a synopsis of the video from this camera captured during the last hour" or "I would like to watch in five minutes a synopsis of the last week," etc. Responding to such a query, the most interesting events ("tubes") are collected from the desired period and are assembled into a synopsis video of the desired length. The synopsis video is an index into the original video as each object includes a pointer to its original time.

While webcam video is endless and the number of objects is unbounded, the available data storage for each webcam may be limited. To keep a finite object queue, we propose a procedure for removing objects from this queue when space is exhausted. Removing objects from the queue

- *Y. Pritch and S. Peleg are with the School of Computer Science and Engineering, The Hebrew University of Jerusalem, 91904 Jerusalem, Israel. E-mail: yaelpri@cs.huji.ac.il, peleg@mail.huji.ac.il.*
- *A. Rav-Acha is with the Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, 76100 Rehovot, Israel. E-mail: ravacha@gmail.com.*

Manuscript received 12 Sept. 2007; revised 16 Dec. 2007; accepted 8 Jan. 2008; published online 23 Jan. 2008.
Recommended for acceptance by J.Z. Wang, D. Geman, J. Luo, and R.M. Gray.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMISI-2007-09-0580.
Digital Object Identifier no. 10.1109/TPAMI.2008.29.

Published by the IEEE Computer Society

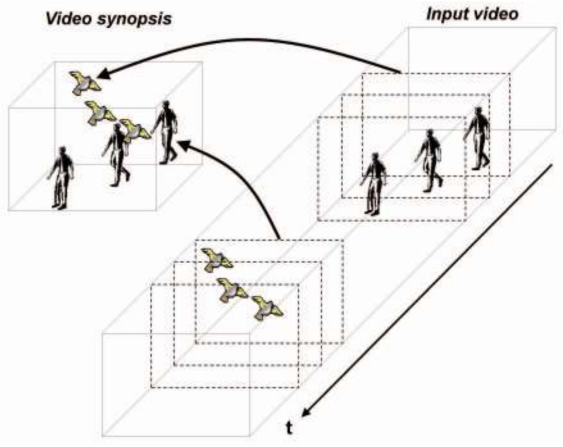

Fig. 1. The input video shows a walking person and, after a period of inactivity, displays a flying bird. A compact video synopsis can be produced by playing the bird and the person simultaneously.

should be done according to similar importance criteria as done when selecting objects for inclusion in the synopsis, allowing the final optimization to examine fewer objects.

In Section 2, a region-based video synopsis is described which produces a synopsis video using optimizations on Markov Random Fields (MRFs) [14]. The energy function in this case consists of low-level costs that can be described by an MRF.

In Section 3, an object-based method for video synopsis is presented. Moving objects are first detected and segmented into space-time "tubes." An energy function is defined on the possible time shifts of these tubes, which encapsulates the desired properties of the video synopsis. This energy function will help to preserve most of the original activity of the video, while avoiding collisions between different shifted activities (tubes). Moving object detection is also done in other object-based video summary methods [13], [10], [31]. However, these methods use object detection to identify significant key frames and do not combine activities from different time intervals.

One of the effects of video synopsis is the display of multiple dynamic appearances of a single object. This effect is a generalization of the "stroboscopic" still pictures used in traditional video synopsis of moving objects [11], [1]. A synopsis can also be generated from a video captured by panning cameras. Stroboscopic and panoramic effects of video synopsis are described in Section 3.4.

The special challenges in creating video synopsis for endless video, such as the ones generated by surveillance cameras, are presented in Section 4. These challenges include handling a varying background due to day-night differences, incorporating an object queue to handle a large amount of objects (Section 4.2), and stitching the synopsis video onto a time-lapse background, as described in Section 4.3. Examples for synopsis of an endless video are given in Section 4.7. The application of video synopsis for indexing is described in Section 1.

Since this work presents a video-to-video transformation, the reader is encouraged to view the video examples at http://www.vision.huji.ac.il/video-synopsis/.

1.1 Related Work on Video Abstraction

A video clip describes visual activities along time and compressing the time axis allows viewing a summary of such a clip in a shorter time. Fast-forward, where several frames are skipped between selected frames, is the most common tool used for video summarization. A special case of fast-forward is called "time lapse," generating a video of very slow processes like growth of flowers, etc. Since fast-forward may lose fast activities during the dropped frames, methods for adaptive fast-forward have been developed [17], [23]. Such methods attempt to skip frames in periods of low interest or lower activity and keep frames in periods of higher interest or higher activity. A similar approach extracts from the video a collection of short video sequences best representing its contents [30]. In [16], different sources for user attention—sound, camera motion, object motion, color, etc.—are discussed. In [33], a survey of fast video browsing is given.

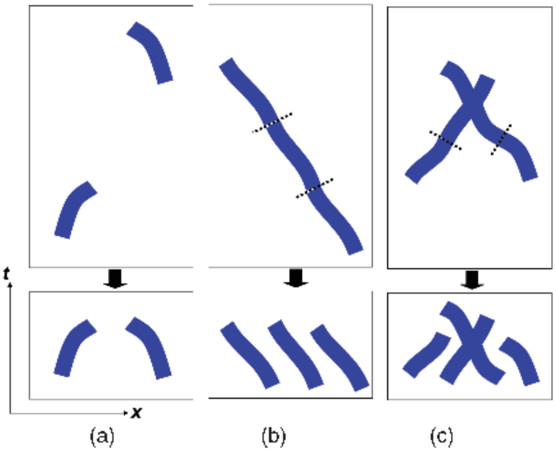

Fig. 2. Schematic description of basic temporal rearrangement of objects. Objects of interest are represented by "activity tubes" in the space-time representation of the video. The upper parts in this figure represent the original video and the lower parts represent the video synopsis. (a) Two objects recorded at different times are shifted to the same time interval in the shorter video synopsis. (b) A single object moving during a long time is broken into segments having a shorter duration and those segments are shifted in time and played simultaneously, creating a dynamic stroboscopic effect. (c) Intersection of objects does not disturb the synopsis when object tubes are broken into segments.

Many approaches to video summary eliminate the time axis and show a synopsis of the video using some key frames [13], [35]. These key frames can be selected arbitrarily or selected according to some importance criteria. But, key frame representation loses the dynamic aspect of video. Comprehensive surveys on video abstraction appear in [15], [18].

In both approaches above, entire frames are used as the fundamental building blocks. A different methodology uses mosaic images together with some metadata for video indexing [11], [24], [21]. In this case, the static synopsis image includes objects from different times.

Object-based approaches to video synopsis were first presented in [26], [12], [25], where moving objects are represented in the space-time domain. These papers introduced a new concept: creating a synopsis video that combines activities from different times (Fig. 1). This paper is a unification and expansion of the approach described in [26], [25].

The underlying idea of the "Video Montage" paper [12] is closely related to ours. In that work, a space-time

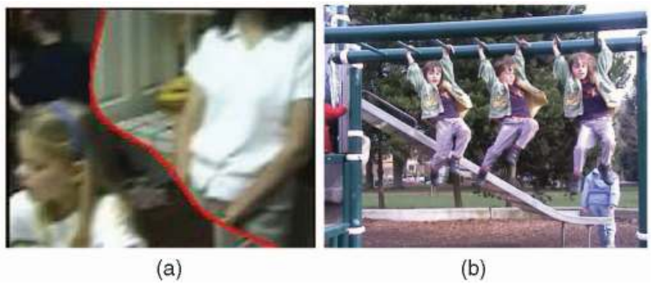

Fig. 3. Comparison between "video montage" [12] and our approach of "video synopsis." (a) A frame from a "video montage." Two space-time regions were shifted in both time and space and then stitched together. Visual seams between the different regions are unavoidable. (b) A frame from a "video synopsis." Only temporal shifts were applied, enabling seamless stitching.

approach for video summarization is presented: Both the spatial and temporal information in a video sequence are simultaneously analyzed and informative space-time portions of the input videos are extracted. Following this analysis, spatial as well as temporal shifts are applied to objects to create a video summary. The basic difference in our paper is the use of only temporal transformations, keeping spatial locations intact. This basic difference results in many differences of object extraction and video composition. Our approach of allowing only temporal transformations prevents the total loss of context that occurs when both the spatial and temporal locations are changing. In addition, maintaining the spatial locations of objects allows the generation of seamless video, avoiding the visually unpleasant seams that appear in the "video montage." These differences are visualized in Fig. 3.

Shifting video regions in time is also done in [29], but for an opposite purpose. In that paper, an infinite video is generated from a short video clip by separating objects (video sprites) from the background and rendering them at arbitrary video locations to create an endless video.

2 Synopsis by Energy Minimization

Let N frames of an input video sequence be represented in a 3D space-time volume $I(x,y,t)$, where (x,y) are the spatial coordinates of the pixel and $1 \leq t \leq N$ is the frame number.

The generated synopsis video $S(x,y,t)$ should have the following properties:

- The video synopsis S should be substantially shorter than the original video I.
- Maximum "activity" (or interest) from the original video should appear in the synopsis video.
- The dynamics of the objects should be preserved in the synopsis video. For example, regular fast-forward may fail to preserve the dynamics of fast objects.
- Visible seams and fragmented objects should be avoided.

The synopsis video S having the above properties is generated with a mapping M, assigning to every coordinate (x,y,t) in the video synopsis S the coordinates of a source pixel from the input video I. We focus in this paper on time shift of pixels, keeping the spatial locations fixed. Thus, any synopsis pixel $S(x,y,t)$ can come from an input pixel $I(x,y,M(x,y,t))$. The time shift M is obtained by minimizing the following cost function:

$$E(M) = E_a(M) + \alpha E_d(M), \tag{1}$$

where $E_a(M)$ (activity) indicates the loss in activity and $E_d(M)$ (discontinuity) indicates the discontinuity across seams having a relative weight of α. The loss of activity will be the number of active pixels in the input video I that do not appear in the synopsis video S, or the weighted sum of their activity measures in the continuous case.

The activity measure of each pixel can be represented by the characteristic function indicating its difference from the background:

$$\chi(x,y,t) = \|I(x,y,t) - B(x,y,t)\|, \tag{2}$$

where $I(x,y,t)$ is the pixel in the input image and $B(x,y,t)$ is the respective pixel in the background image. To obtain the background image, we can use a temporal median over the entire video. More sophisticated background construction methods can also be used, such as described in [8].

Accordingly, the activity loss is given by

$$E_a(M) = \sum_{(x,y,t)\in I} \chi(x,y,t) - \sum_{(x,y,t)\in S} \chi(x,y,M(x,y,t)). \tag{3}$$

The discontinuity cost E_d is defined as the sum of color differences across seams between spatiotemporal neighbors in the synopsis video and the corresponding neighbors in the input video (a similar formulation can be found in [1]):

$$\begin{aligned} E_d(M) = \sum_{(x,y,t)\in S} \sum_i \| & S((x,y,t)+e_i) \\ & - I((x,y,M(x,y,t))+e_i)\|^2, \end{aligned} \tag{4}$$

where e_i are the six unit vectors representing the six spatiotemporal neighbors: four spatial neighbors and two temporal neighbors. A demonstration of the space-time operations that create a short video synopsis by minimizing the cost function (1) is shown in Fig. 4a.

2.1 Minimization of the Energy Function

Notice that the cost function $E(M)$ (1) corresponds to a 3D MRF, where each node corresponds to a pixel in the 3D volume of the output movie and can be assigned any time value corresponding to an input frame. The weights on the nodes are determined by the activity cost, while the edges between nodes are determined according to the discontinuity cost. The cost function can therefore be minimized by algorithms like iterative graph cuts [14].

The optimization of (1), allowing each pixel in the video synopsis to come from any time, is a difficult problem. For example, an input video of 3 minutes which is summarized into a video synopsis of 5 seconds results in a graph of 2^{25} nodes (5 seconds × 30 frames/seconds × image size of 640 × 480), each having 5,400 possible labels (3 minutes × 60 seconds × 30 frames/seconds).

It was shown in [2] that, for cases of dynamic textures or objects that move in a horizontal path, 3D MRFs can be solved efficiently by reducing the problem into a 1D problem. In this work, we address objects that move in a more general way and, therefore, we use different constraints. Consecutive pixels in the synopsis video S are restricted to coming from consecutive pixels in the input video I. Under this restriction, the 3D graph is reduced to a 2D graph, where each node corresponds to a spatial location in the synopsis movie. The label of each node $M(x,y)$ determines the frame number t in I

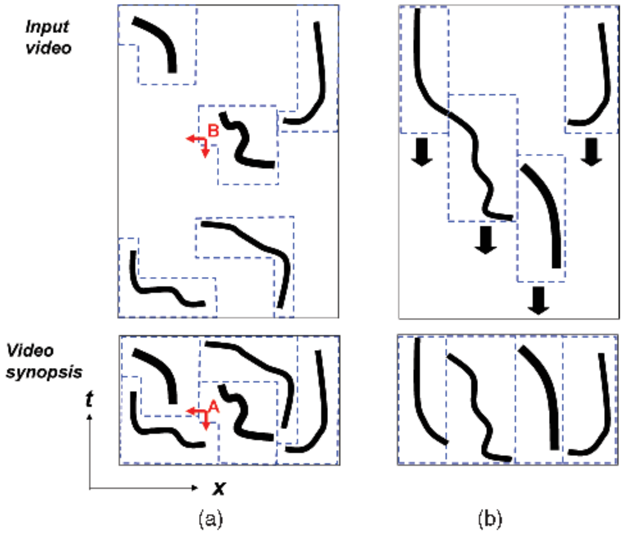

Fig. 4. In this space-time representation of video, moving objects create the "activity tubes." The upper part represents the original video I, while the lower part represents the video synopsis S. (a) The shorter video synopsis S is generated from the input video I by including most active pixels together with their spatiotemporal neighborhood. To assure smoothness, when pixel A in S corresponds to pixel B in I, their "cross border" neighbors in space as well as in time should be similar. (b) An approximate solution can be obtained by restricting consecutive synopsis pixels to come from consecutive input pixels.

shown in the first frame of S, as illustrated in Fig. 4b. A seam exists between two neighboring locations (x_1, y_1) and (x_2, y_2) in S if $M(x_1, y_1) \neq M(x_2, y_2)$ and the discontinuity cost $E_d(M)$ along the seam is a sum of the color differences at this spatial location over all frames in S:

$$E_d(M) = \sum_{x,y} \sum_{i} \sum_{t=1}^{K} \| S((x,y,t) + e_i) - I((x,y,M(x,y)+t) + e_i) \|^2, \tag{5}$$

where e_i are now four unit vectors describing the four spatial neighbors.

The number of labels for each node is $N - K$, where N and K are the number of frames in the input and output videos, respectively. The activity loss for each pixel is

$$E_a(M) = \sum_{x,y} \left(\sum_{t=1}^{N} \chi(x,y,t) - \sum_{t=1}^{K} \chi(x,y,M(x,y)+t) \right).$$

Fig. 5 shows an original frame and a frame from a synopsis video that was obtained using this approximation.

To overcome the computational limitations of the region-based approach and to allow the use of higher level cost functions, an object-based approach for video synopsis is proposed. This object-based approach is described in the following section and will also be used for handling endless videos from webcams and surveillance cameras.

3 Object-Based Synopsis

The low-level approach for video synopsis as described earlier is limited to satisfying local properties such as avoiding visible seams. Higher level object-based properties can be incorporated when objects can be detected and tracked. For example, avoiding the stroboscopic effect requires the detection and tracking of each object in the volume. This section describes an implementation of an object-based approach for video synopsis. Several object-based video summary methods exist in the literature (for example, [13], [10], [31]) and they all use the detected objects for the selection of significant frames. Unlike these methods, we shift objects in time and create new synopsis frames that never appeared in the input sequence in order to make better use of space and time.

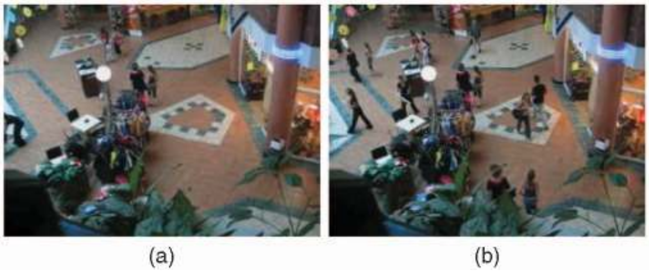

Fig. 5. The activity in a surveillance video can be condensed into a much shorter video synopsis. (a) A typical frame from the original video taken in a shopping mall. (b) A frame from the video synopsis.

3.1 Object Detection and Segmentation

In order to generate a useful synopsis, interesting objects and activities (*tubes*) should be identified. In many cases, the indication of interest is simple: A moving object is interesting. While we use object motion as an indication of interest in many examples, exceptions must be noted. Some motions may have little importance, like leaves on a tree or clouds in the sky. People or other large animals in the scene may be important even when they are not moving. While we do not address these exceptions, it is possible to incorporate object recognition (e.g., people detection [19], [22]), dynamic textures [9], or detection of unusual activities [5], [34]. We will give a simple example of video synopsis giving preferences to different classes of objects.

As objects are represented by tubes in the space-time volume, we use the words "objects" and "tubes" interchangeably.

To enable segmentation of moving foreground objects, we start with background construction. In short video clips, the appearance of the background does not change and it can be built by using a temporal median over the entire clip. In the case of surveillance cameras, the appearance of the background changes in time due to changes in lighting, changes of background objects, etc. In this case, the background for each time can be computed using a temporal median over a few minutes before and after each frame. We normally use a median over 4 minutes. Other methods for background construction are possible, even when using a shorter temporal window [8], but we used the median due to its efficiency. Fig. 6 shows several background images from a surveillance video as they vary during the day.

We use a simplification of [32] to compute the space-time tubes representing dynamic objects. Background subtraction is combined together with min-cut to get smooth segmentation of foreground objects. As in [32], image gradients that coincide with background gradients are attenuated as they are less likely to be related to motion boundaries. The resulting "tubes" are connected

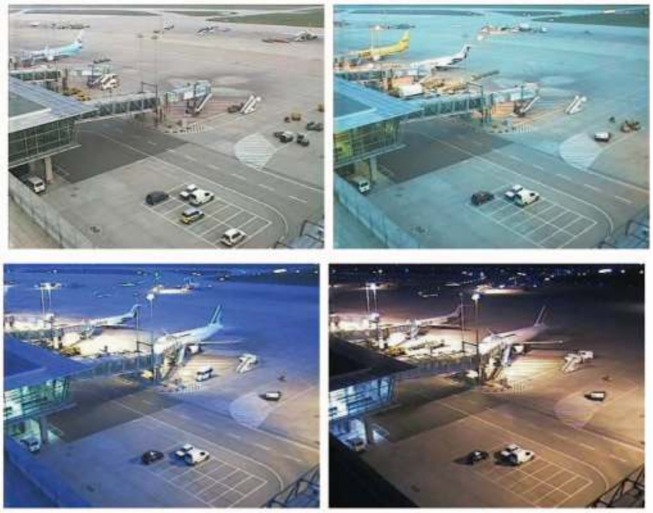

Fig. 6. Background images from a surveillance camera at Stuttgart airport. The bottom images are at night, while the top images are in daylight. Parked cars and parked airplanes become part of the background. This figure is best viewed in color.

components in the 3D space-time volume and their generation is briefly described below.

Let B be the current background image and let I be the current image to be processed. Let V be the set of all pixels in I and let N be the set of all adjacent pixel pairs in I. A labeling function f labels each pixel r in the image as foreground ($f_r = 1$) or background ($f_r = 0$). A desirable labeling f usually minimizes the Gibbs energy [6]:

$$E(f) = \sum_{r \in V} E_1(f_r) + \lambda \sum_{(r,s) \in N} E_2(f_r, f_s), \tag{6}$$

where $E_1(f_r)$ is the unary-color term, $E_2(f_r, f_s)$ is the pairwise-contrast term between adjacent pixels r and s, and λ is a user-defined weight.

As a pairwise-contrast term, we used the formula suggested by [32]:

$$E_2(f_r, f_s) = \delta(f_r - f_s) \cdot \exp(-\beta d_{rs}), \tag{7}$$

where $\beta = 2 < \|(I(r) - I(s)\|^2 >^{-1}$ is a weighting factor ($< \cdot >$ is the expectation over the image samples) and d_{rs} are the image gradients, attenuated by the background gradients, and given by

$$d_{rs} = \|(I(r) - I(s)\|^2 \cdot \frac{1}{1 + \left(\frac{\|B(r)-B(s)\|}{K}\right)^2 \exp(\frac{-z_{rs}^2}{\sigma_z})}. \tag{8}$$

In this equation, z_{rs} measures the dissimilarity between the foreground and the background,

$$z_{rs} = \max \|I(r) - B(r)\|, \|I(s) - B(s)\|, \tag{9}$$

and K and σ_z are parameters, set to 5 and 10, respectively, as suggested by [32].

As for the unary-color term, let $d_r = \|I(r) - B(r)\|$ be the color differences between the image I and the current background B. The foreground (1) and background (0) costs for a pixel r are set to

$$\begin{aligned} E_1(1) &= \begin{cases} 0 & d_r > k_1 \\ k_1 - d_r & otherwise, \end{cases} \\ E_1(0) &= \begin{cases} \infty & d_r > k_2, \\ d_r - k_1, & k_2 > d_r > k_1 \\ 0 & otherwise, \end{cases} \end{aligned} \tag{10}$$

where k_1 and k_2 are user-defined thresholds. Empirically, $k_1 = 30/255$ and $k_2 = 60/255$ worked well in our examples.

We do not use a lower threshold with infinite weights since the later stages of our algorithm can robustly handle pixels that are wrongly identified as foreground, but not the opposite. For the same reason, we construct a mask of all foreground pixels in the space-time volume and apply a 3D morphological dilation on this mask. As a result, each object is surrounded by several pixels from the background. This fact will be used later by the stitching algorithm.

Finally, the 3D mask is grouped into connected components, denoted as "activity tubes." Examples of extracted tubes are shown in Figs. 7 and 8.

Each tube b is represented by its characteristic function

$$\chi_b(x, y, t) = \begin{cases} \|I(x, y, t) - B(x, y, t)\| & t \in t_b \\ 0 & otherwise, \end{cases} \tag{11}$$

where $B(x, y, t)$ is a pixel in the background image, $I(x, y, t)$ is the respective pixel in the input image, and t_b is the time interval in which this object exists.

3.2 Energy between Tubes

In this section, we define the energy of interaction between tubes. This energy will later be used by the optimization stage, creating a synopsis having maximum activity while avoiding conflicts and overlap between objects. Let B be the set of all activity tubes. Each tube b is defined over a finite time segment in the original video stream $t_b = \left[t_b^s, t_b^e\right]$.

The synopsis video is generated based on a temporal mapping M, shifting objects b in time from its original time in the input video into the time segment $\hat{t_b} = \left[\hat{t_b^s}, \hat{t_b^e}\right]$ in the video synopsis. $M(b) = \hat{b}$ indicates the time shift of tube b into the synopsis and, when b is not mapped to the output synopsis, $M(b) = \emptyset$. We define an optimal synopsis video as the one that minimizes the following energy function:

$$E(M) = \sum_{b \in B} E_a(\hat{b}) + \sum_{b,b' \in B} \Big(\alpha E_t(\hat{b}, \hat{b}') + \beta E_c(\hat{b}, \hat{b}')\Big), \tag{12}$$

where E_a is the activity cost, E_t is the temporal consistency cost, and E_c is the collision cost, all defined below. Weights α and β are set by the user according to their relative importance for a particular query. Reducing the weights of the collision cost, for example, will result in a denser video where objects may overlap. Increasing this weight will result in a sparser video where objects do not overlap and less activity is presented. An example for the different synopsis obtained by varying β is given in Fig. 16.

Note that the object-based energy function in (12) is different from the low-level energy function defined in (1). After extracting the activity tubes, the pixel-based cost can be replaced with object-based cost. Specifically, the Stitching cost in (1) is replaced by the Collision cost in (12) (described next). This cost penalizes for stitching two different objects together, even if their appearance is similar (e.g., two people). In addition, a "Temporal Consistency"

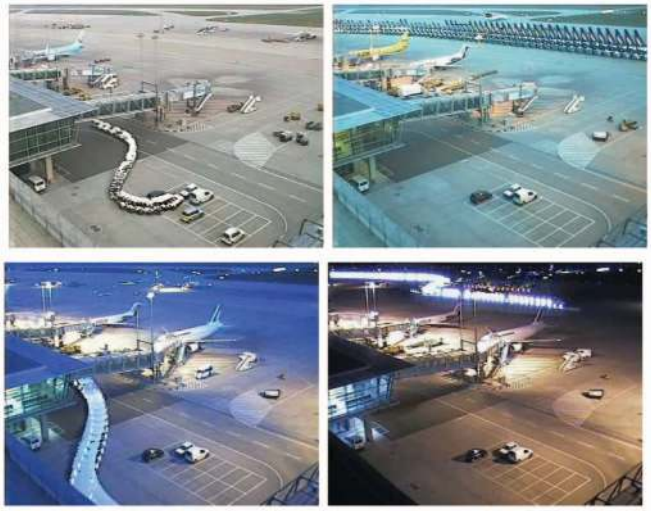

Fig. 7. Four extracted tubes shown "flattened" over the corresponding backgrounds from Fig. 6. The left tubes correspond to ground vehicles, while the right tubes correspond to airplanes on the runway at the back. This figure is best viewed in color.

cost is defined, penalizing for the violation of the temporal relations between objects (or tubes). Such features of the synopsis are harder to express in terms of pixel-based costs.

3.2.1 Activity Cost

The activity cost favors synopsis movies with maximum activity. It penalizes for objects that are not mapped to a valid time in the synopsis. When a tube is excluded from the synopsis, i.e., $M(b) = \emptyset$, then

$$E_a(\hat{b}) = \sum_{x,y,t} \chi_{\hat{b}}(x,y,t), \tag{13}$$

where $\chi_b(x,y,t)$ is the characteristic function as defined in (11). For each tube b whose mapping $\hat{b} = M(b)$ is partially included in the final synopsis, we define the activity cost similar to (13), but only pixels that were not entered into the synopsis are added to the activity cost.

3.2.2 Collision Cost

For every two "shifted" tubes and every relative time shift between them, we define the collision cost as the volume of their space-time overlap weighted by their activity measures:

$$E_c(\hat{b},\hat{b'}) = \sum_{x,y,t\in\hat{t_b}\cap\hat{t_{b'}}} \chi_{\hat{b}}(x,y,t)\chi_{\hat{b'}}(x,y,t), \tag{14}$$

where $\hat{t_b}\cap\hat{t_{b'}}$ is the time intersection of b and b' in the synopsis video. This expression will give a low penalty to pixel whose color is similar to the background but was added to an activity tube in the morphological dilation process. Changing the weight of the collision cost E_c changes the density of objects in the synopsis video, as shown in Fig. 16.

3.2.3 Temporal Consistency Cost

The temporal consistency cost adds a bias toward preserving the chronological order of events. The preservation of chronological order is more important for tubes that have a strong interaction. For example, it would be preferred to keep the relative time of two people talking to each other or to keep the chronological order of two events with a reasoning relation. Yet, it is very difficult to detect such interactions. Instead, the amount of interaction $d(b,b')$ between each pair of tubes is estimated from their relative spatiotemporal distance, as described below:

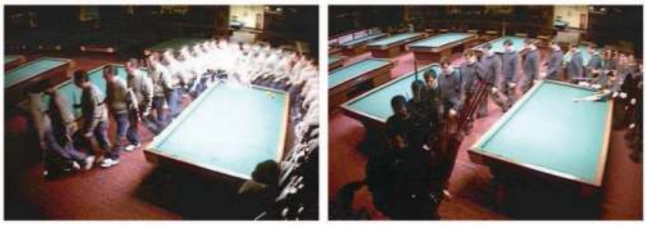

Fig. 8. Two extracted tubes from the "Billiard" scene.

$$\begin{aligned} &\text{if } \hat{t_b}\cap\hat{t_{b'}} \neq \emptyset \quad \text{then} \\ &d(b,b') = \exp\Big(-\min\nolimits_{t\in\hat{t_b}\cap\hat{t_{b'}}}\{d(b,b',t)\}/\sigma_{space}\Big), \end{aligned} \tag{15}$$

where $d(b,b',t)$ is the euclidean distance between the pair of closest active pixels from b and b' in frame t and σ_{space} determines the extent of the space interaction between tubes.

If tubes b and b' do not share a common time at the synopsis video and assuming that b is mapped to earlier time than b', their interaction diminishes exponentially with time:

$$d(b,b') = \exp\big(-(\hat{t}^s_{b'} - \hat{t}^e_b)/\sigma_{time}\big), \tag{16}$$

where σ_{time} is a parameter defining the extent of time in which events still have temporal interaction.

The temporal consistency cost creates a preference for maintaining the temporal relations between objects by penalizing cases where these relations are violated:

$$E_t(\hat{b},\hat{b'}) = d(b,b')\cdot\begin{cases} 0 & t^s_{b'} - t^s_b = \hat{t}^s_{b'} - \hat{t}^s_b \\ C & \text{otherwise,} \end{cases} \tag{17}$$

where C is a constant penalty for events that do not preserve temporal consistency.

3.3 Energy Minimization

Since the global energy function in (12) (and later in (20)) is written as a sum of energy terms defined on single tubes or pairs of tubes, it can be minimized by various MRF-based techniques such as Belief Propagation or Graph Cuts [14]. We used a simple greedy optimization, which gave good results. The optimization was applied in the space of all possible temporal mappings M, including the special case when a tube is not used at all in the synopsis video.

Each state describes the subsets of tubes that are included in the synopsis and their mapping into the synopsis. Neighboring states are defined as states in which a single activity tube is removed or changes its mapping into the synopsis. As an initial state, we used the state in which all tubes are shifted to the beginning of the synopsis movie. Also, in order to accelerate computation, we restricted the temporal shifts of tubes to be in jumps of 10 frames.

3.4 Stroboscopic Panoramic Synopsis

When long tubes exist in the input video, no temporal rearrangement of the tubes can give a very short video as

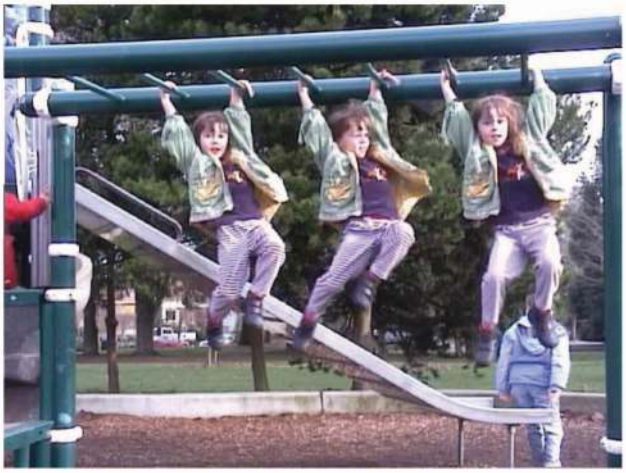

Fig. 9. Video synopsis with the dynamic stroboscopic effect as illustrated schematically in Fig. 2b. The video can be seen at http://www.vision.huji.ac.il/video-synopsis.

the duration of the synopsis video is bounded from below by the duration of the longest tube that is shown in the synopsis. There are a few options to overcome this limitation: One option is to display only partial activities (i.e., to allow displaying subsections of a tube). Another option is to cut the long activity tube into shorter subsections and display several subsections simultaneously. This results in a dynamic stroboscopic effect—simultaneously displaying several appearances of the same object. This effect is described schematically in Fig. 2b and an example appears in Fig. 9.

An example where the stroboscopic effect is very useful is in the case of a panning video camera scanning a scene. In this case, spatial redundancy can be eliminated by using a panoramic mosaic. Yet, existing methods construct a single panoramic image in which the scene dynamics is lost. Dynamics has been represented by a static stroboscopic image [11], [1], [4], where moving objects are displayed at several locations along their paths.

A panoramic synopsis video can be created from a panning camera by simultaneously displaying actions that took place at different times in different regions of the scene. The duration of a panoramic video is limited by the duration of time each object is visible by the camera. In the special case of a camera tracking an object, the time duration of the tracked object equals the time duration of the entire video. Temporal compression can be achieved only by allowing the stroboscopic effect, as shown schematically in Fig. 10. An example of a camera tracking a running leopard is shown in Fig. 11.

Constructing the panoramic video synopsis is done in a similar manner to the regular video synopsis, with a preliminary stage of aligning all the frames to a reference frame.

3.5 Surveillance Applications

An interesting application for video synopsis may be access to stored surveillance videos. When it becomes necessary to examine certain events in the video, it can be done much faster with video synopsis. Two examples are given from real surveillance cameras. Fig. 12 uses a video captured by a camera watching a city street, with pedestrians occasionally crossing the field of view. Many of them can be collected into a very condensed synopsis.

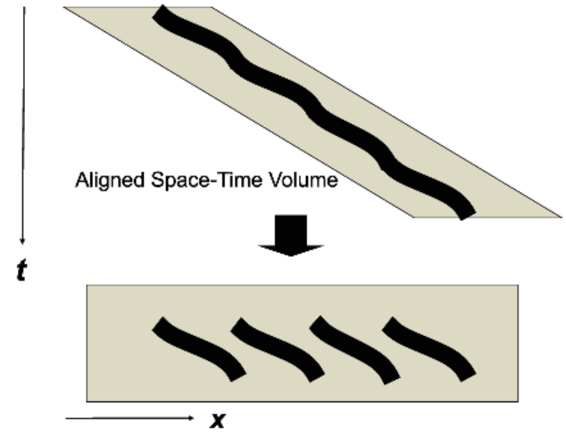

Fig. 10. A schematic diagram of panoramic video synopsis. When the frames of a panning video are aligned, the obtained space-time volume is titled according to the motion of the camera. The long tube in the center of the space-time volume represents a tracked object which is tracked by the panning camera. The background of the panoramic video synopsis, whose space-time volume is at the bottom, is a mosaic of the background. The long input tube is broken into shorter segments that are shifted into the space-time volume of the video synopsis. Each output frame will simultaneously show several occurrences of the tracked object.

4 Synopsis of Endless Video

As mentioned earlier, millions of webcams and surveillance cameras are covering the world, capturing their field of view 24 hours/day. One of the problems in utilizing these cameras is that they provide unedited raw data. A 2 hour feature film, for example, is usually created from hundreds or even thousands of hours of raw video footage. Without editing, most of the webcam data is irrelevant. Also, viewing a camera on another continent may be convenient only during hours of nonactivity because of time zone differences.

In this section, we attempt to make the webcam resource more useful by giving the viewer the ability to view summaries of the endless video, in addition to the live video stream provided by the camera. A user's query can be similar to "I would like to watch in five minutes a synopsis of last week." To enable this, we describe a system that is based on the object-based synopsis, but which consists of additional components that allows dealing with endless videos.

In this system, a server can view the live video feed, analyze the video for interesting events, and record an object-based description of the video. This description lists, for each camera, the interesting objects, their duration, their location, and their appearance. In a 3D space-time description of the video, each object is a "tube."

A two-phase process is proposed for synopsis of endless video, as shown in Fig. 13:

1. **Online Phase** during video capture. This phase is done in real time:
   - Creating a background video by temporal median.
   - Object (tube) detection and segmentation (Section 3.1).

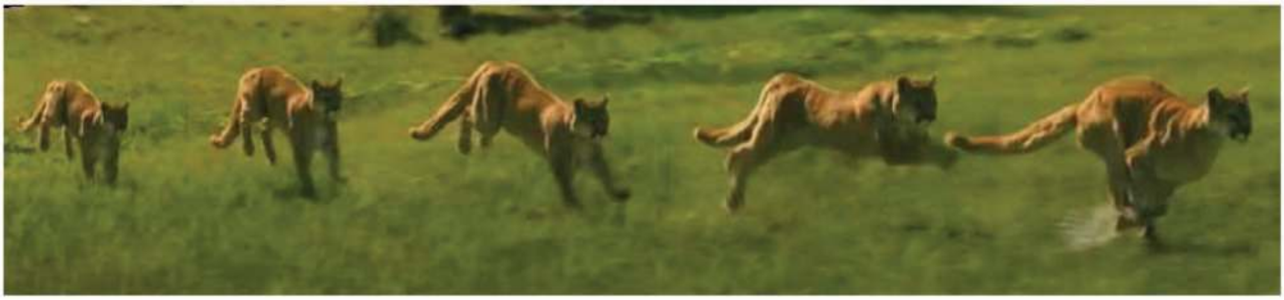

Fig. 11. In this example, a video camera tracks a running leopard. The background of the synopsis video is a panoramic mosaic of the background and the foreground includes several dynamic copies of the running leopard moving simultaneously.

  - Inserting detected objects into the object queue (Section 4.2).
  - Removing objects from the object queue when reaching a space limit (Section 4.2).
2. **Response Phase** constructing a synopsis according to a user query. This phase may take a few minutes, depending on the amount of activity in the time period of interest. This phase includes the following:
  - Constructing a time-lapse video of the changing background (Section 4.4). Background changes are usually caused by day-night differences, but can also be a result of an object that starts (stops) moving.
  - Selecting tubes that will be included in the synopsis video and computing the optimal temporal arrangement of these tubes (Sections 3.2 and 3.3).
  - Stitching the tubes and the background into a coherent video (Section 4.6). This step should take into account that activities from different times can appear simultaneously and on a background from yet another time.

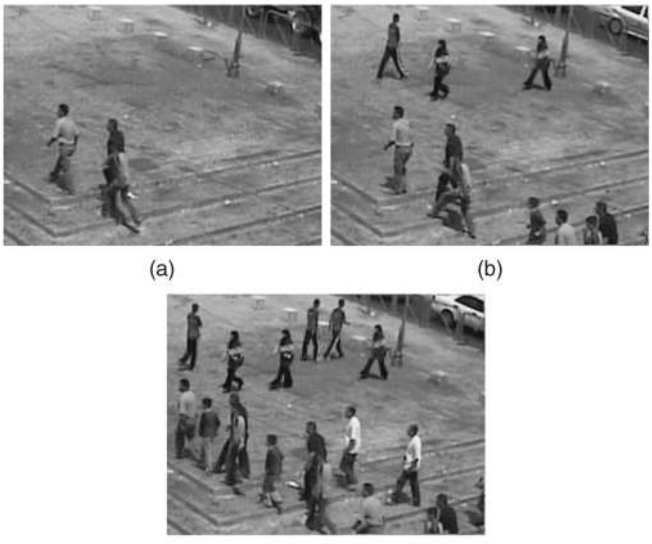

Fig. 12. Video synopsis from street surveillance. (a) A typical frame from the original video (22 seconds). (b) A frame from a video synopsis movie (2 seconds) showing condensed activity. (c) A frame from a shorter video synopsis (0.7 second) showing even more condensed activity.

4.1 Removing Stationary Frames

Most surveillance cameras and webcams have long periods with no activity. The frames corresponding to such time periods can be filtered out during the online phase. The original time of the remaining frames is recorded together with each frame. In our implementation, we recorded frames according to two criteria: 1) a global change in the scene, measured by the sum of squared difference (SSD) between the incoming frame and the last kept frame. This criterion tracked the lighting changes expressed by a gradual illumination change in the entire frame. 2) The existence of a moving object measured by the maximal SSD in small windows.

By assuming that moving objects with a very small duration (e.g., less than a second) are not important, video activity can be measured only once in every 10 frames.

4.2 The Object Queue

One of the main challenges in handling endless videos is developing a scheme to "forget" older objects when new objects arrive. The naive scheme of throwing out the oldest activity is not good as a user may wish to get a summary of a long time duration which includes objects from the entire period. Instead, we propose an alternative scheme that aims at estimating the importance of each object to possible future queries and throwing objects out accordingly.

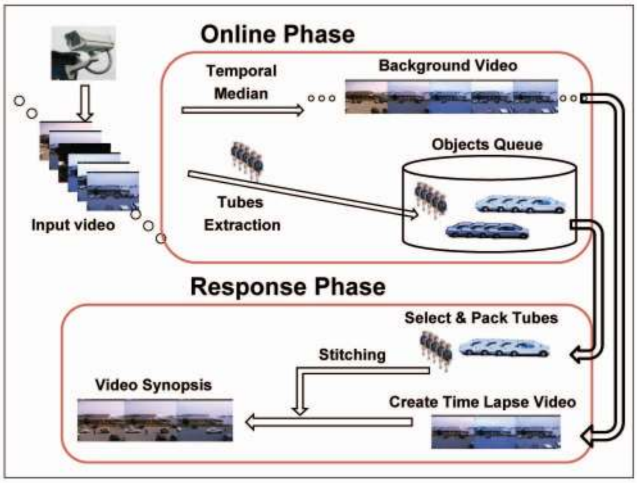

Fig. 13. The two-phase process for creating a synopsis of an endless video. The online phase is performed in real time during video capture and recording. The response phase is performed following a user query and generates the video synopsis.

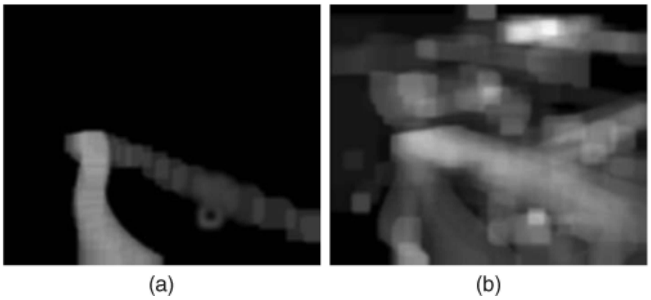

Fig. 14. The spatial distribution of activity in the airport scene (intensity is log of activity value). The activity distribution of a single tube is on the left and the average over all tubes is on the right. As expected, the highest activity is on the car lanes and on the runway. The potential for the collision of tubes is higher in regions having a higher activity.

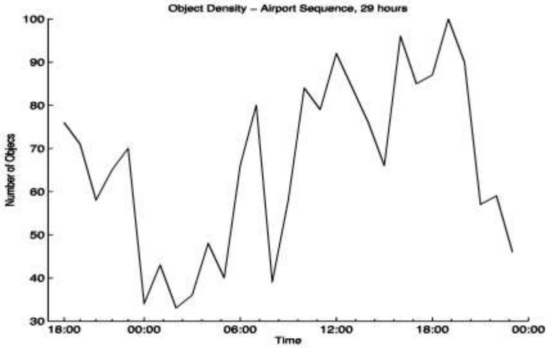

Fig. 15. Temporal distribution of activities, as measured by the number of moving objects, at the airport scene over 29 hours. There are 1,920 objects during this period.

All detected objects, represented as tubes in the space-time volume, are stored in a queue awaiting user queries. When an object is inserted into the queue, its activity cost (13) is computed to accelerate the future construction of synopsis videos. As the video generated by the webcam is endless, it is likely that, at some point, the allocated space will be exhausted and objects will have to be removed from the queue.

When removing objects (tubes) from the queue, we prefer to remove objects that are least likely to be included in a final synopsis. In our examples, we used three simple criteria that can be computed efficiently: "importance" (activity), "collision potential," and "age." But other options are possible, for example, when a specific appearance or activity is of interest.

A possible measure of the importance of an object is the sum of its characteristic function, as defined in (13).

Since the collision cost cannot be computed before receiving the user query, an estimate for the collision cost of tubes is made using the spatial activity distribution in the scene. This spatial activity is represented by an image which is the sum of active pixels of all objects in each spatial location, normalized to sum to one. A similar spatial activity distribution is computed for each individual object (this time unnormalized). The correlation between these two activity distributions is used as a "potential collision" cost for this object. An image showing the activity distribution in a scene is shown in Fig. 14.

There are several possible approaches to addressing the removal of older objects from the queue, taking into consideration the desired distribution of objects in the synopsis. For example, the user can be interested in focusing on newer events but leaving some representation of old events in case they were significant. Alternatively, the synopsis should have a uniform representation of every time interval (e.g., in a synopsis of 24 hours, a user may be interested in seeing an object from each and every hour if applicable).

In the first approach, we can assume that the density of objects in the queue should decrease exponentially with the age of the objects. For example, if we divide the age axis into discrete time intervals, the number of objects at t's interval, N_t, should be proportional to

$$N_t = K\frac{1}{\sigma}e^{-\frac{t}{\sigma}}, \tag{18}$$

where σ is the decay coefficient and K is determined to control the total number of objects in the queue. When an object should be removed from the queue, the number of objects in each time interval t is compared to N_t. Only objects from time intervals t whose population exceeds N_t will be evaluated using the activity cost and the potential collision. The object with minimal activity and maximal collision will be removed.

An example of the temporal distribution of the object arriving into the queue appears in Fig. 15. Exponential decay of objects in the queue will result in an age distribution which is proportional to the arrival distribution multiplied by a decaying exponential.

4.3 Synopsis Generation

The object queue can be accessed via queries such as "I would like to have a one-minute synopsis of this camera broadcast during the past day." Given the desired period from the input video and the desired length of the synopsis, the synopsis video is generated using four steps:

1. Generating a background video.
2. Once the background video is defined, a consistency cost is computed for each object and for each possible time in the synopsis.
3. An energy minimization step determines which tubes (space-time objects) appear in the synopsis and at what time.
4. The selected tubes are combined with the background time-lapse to get the final synopsis.

These steps are described in this section. The reduction of the original video to an object-based representation enables a fast response to queries.

After a user query about a second (shorter) time period, a queue is generated having only objects from the desired time period. To enable fast optimization, the collision cost in (14) between every two objects in the smaller queue is computed in advance.

4.4 Time-Lapse Background

The background of the synopsis video is a time-lapse background video, generated before adding activity tubes into the synopsis. The background video has two tasks: 1) It should represent the background changes over time (e.g., day-night transitions, etc.) and 2) it should represent the background of the activity tubes. These two goals are conflicting as representing the background of activity tubes will be done best when the background video covers only active periods, ignoring, for example, most night hours.

We address this trade-off by constructing two temporal histograms: 1) a temporal activity histogram H_a of the video stream (an example of such histogram is shown in Fig. 15) and 2) a uniform temporal histogram H_t. We compute a third histogram by interpolating the two histograms $\lambda \cdot H_a + (1-\lambda) \cdot H_t$, where λ is a weight given by the user. With $\lambda = 0$, the background time-lapse video will be uniform in time regardless of the activities, while, with $\lambda = 1$, the background time-lapse video will include the background only from active periods. We usually use $0.25 < \lambda < 0.5$.

Background frames are selected for the time-lapse background video according to the interpolated temporal histogram. This selection is done such that the area of the histogram between every two selected background frames is equal. More frames are selected from active time durations while not totally neglecting inactive periods.

4.5 Consistency with Background

Since we do not assume accurate segmentation of moving objects, we prefer to stitch tubes to background images having a similar appearance. This tube to background consistency can be taken into account by adding a new energy term $E_s(b)$. This term will measure the cost of stitching an object to the time-lapse background. Formally, let $I_{\hat{b}}(x,y,t)$ be the color values of the mapped tube $\hat{b}$ and let $B_{out}(x,y,t)$ be the color values of the time-lapse background. We set

$$E_s\left(\hat{b}\right) = \sum_{x,y \in \sigma(\hat{b}), t \in \hat{t}_b \cap t_{out}} \left\| I_{\hat{b}}(x,y,t) - B_{out}(x,y,t) \right\|, \tag{19}$$

where $\sigma(\hat{b})$ is the set of pixels in the border of the mapped activity tube $\hat{b}$ and t_{out} is the duration of the output synopsis. This cost assumes that each tube is surrounded by pixels from its original background (resulting from our morphological dilation of the activity masks).

The background consistency term in (19) is added to the energy function described in (12), giving

$$\begin{aligned} E(M) = & \sum_{b \in B} \left(E_a\left(\hat{b}\right) + \gamma E_s\left(\hat{b}\right) \right) \\ & + \sum_{b,b' \in B} \left(\alpha E_t\left(\hat{b}, \hat{b}'\right) + \beta E_c\left(\hat{b}, \hat{b}'\right) \right), \end{aligned} \tag{20}$$

where α, β, γ are user selected weights that are query dependent. The effect of changing the value of β can be seen in Fig. 16.

4.6 Stitching the Synopsis Video

The stitching of tubes from different time periods poses a challenge to existing methods (such as [1], [20]). Stitching all of the tubes at once may result in a blending of colors from different objects, which is an undesired effect. It is better to preserve the sharp transitions between different objects while eliminating the seams only between the objects and the background. An accurate segmentation of the objects may solve this problem, but an accurate segmentation is unrealistic. Instead, the boundaries of each tube consist of background pixels due to the morphological dilation we apply when generating the activity tubes.

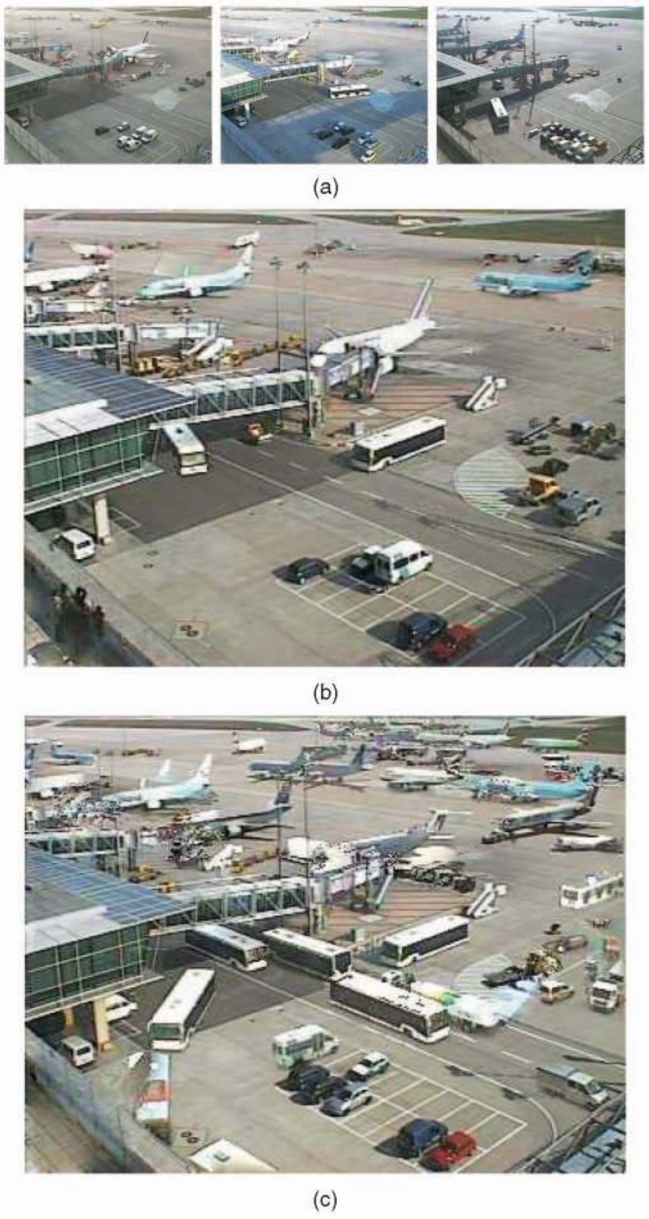

Fig. 16. (a) Three frames from a video captured over 24 hours at Stuttgart airport. (b) A frame from a 20 second synopsis of this period. (c) Reducing the "collision penalty" in the cost function substantially increases the density of objects, allowing more overlap between objects.

The α-Poisson Image Blending proposed in [28] may be a good solution for the stitching between objects, but not as good as Poisson Editing [20] for stitching the objects to the background. The suggested approach is to use the observation that all objects have a similar background (up to illumination changes) and stitch each tube independently to the time-lapse background. Any blending method is possible and, in our experiments, we used a modification of Poisson editing: We add a regularization that preserves the original appearance of the objects even if they were stitched to background images with different lighting conditions (e.g., people seen during the day stitched on top of an evening background).

Let Ω be an image domain with boundary $\partial\Omega$. Let f, b be the foreground object (tube) and background (time-lapse) pixel colors and let s be the unknown values of the stitched

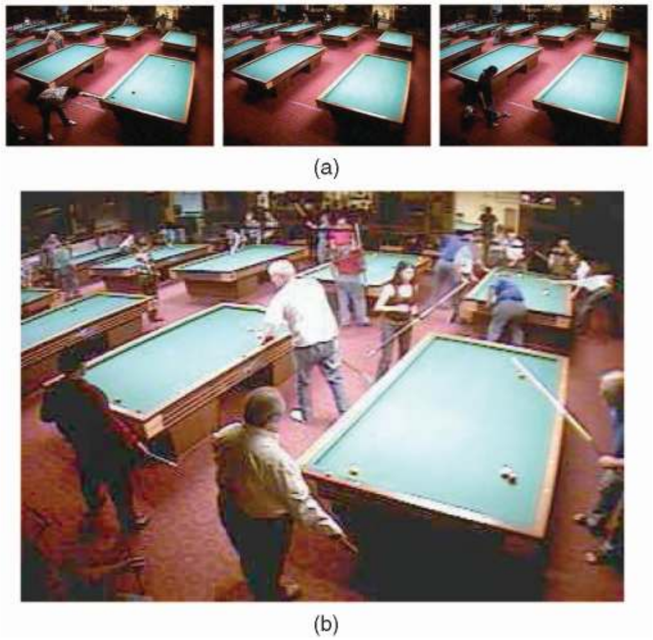

Fig. 17. Top: Three frames from a video captured over 9 hours in a billiards club. Bottom: A frame from a short synopsis of this period. Notice the multiple players per table in the synopsis.

object over the interior of Ω. The result of the Poisson blending with regularization is given by

$$min_s \sum_{\Omega} \Big[(\Delta s - \Delta f)^2 + \lambda (s - f).^2 \Big] \quad such\ that \quad s_{\partial\Omega} = b_{\partial\Omega},$$

where λ is the weight of the regularization term. In [3], it was shown that gradient domain stitching can be very efficient.

After stitching each tube to the background, overlapping tubes are blended together by letting each pixel be a weighted average of the corresponding pixels from the stitched activity tubes $\hat{b}$, with weights proportional to the activity measures $\chi_{\hat{b}}(x, y, t)$. Alternatively, transparency can be avoided by taking the pixel with maximal activity measure instead of the weighted average.

It may be possible to use depth ordering when "object tubes" are combined, where closer tubes will occlude further tubes. A simple "ground plane" heuristic can be used which assumes that an object whose vertical image position is lower is also closer. Other depth ordering methods include [7]. The frequency of object occlusion cases depends on the relative weights of the collision cost (that prevent such cases) with respect to other costs.

4.7 Examples

We tested video synopsis on a few video streams captured off the Internet. As the frame rate is not constant over the Internet and frames drop periodically, whenever we use a temporal neighborhood, we do not count the number of frames, but we use the absolute times of each frame.

Figs. 16 and 18 are from cameras stationed outdoors, while Fig. 17 is from a camera stationed indoors with constant lighting. In most examples, the main "interest" of each tube has been the number of moving pixels in it.

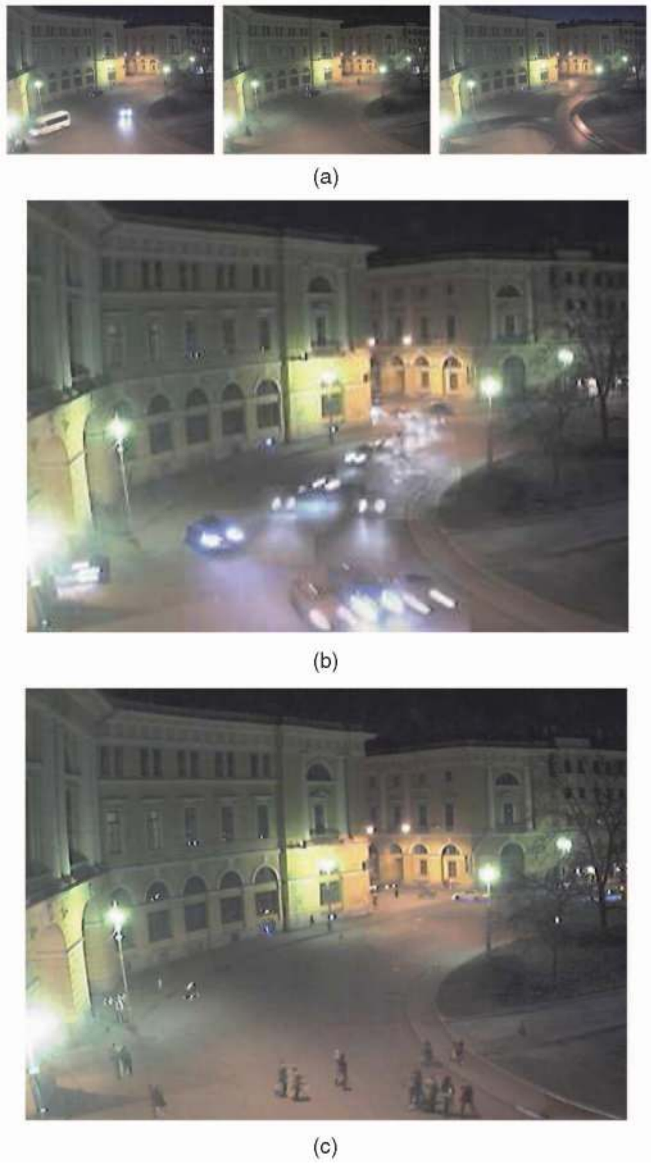

Fig. 18. (a) Three frames from a video captured overnight in St. Petersburg. The street had little activity. (b) A frame from a short synopsis of this video. Cars that passed during different hours are shown simultaneously. (c) Selecting only small, dark objects creates a new synopsis video—now with people rather than with cars.

Fig. 16 shows the effect of the choice of collision cost of the density of objects in the video synopsis. Fig. 18 shows shape-based preferences. In Fig. 18b, the regular cost function was used and the large objects (moving cars) were preferred. In Fig. 18c, small, dark objects were preferred, showing a completely different pedestrian activity.

5 Applications for Video Synopsis

The proposed video synopsis is a general framework that can be adopted for multiple applications. Some variants of this framework are described in this section.

5.1 Video Indexing

Video synopsis can be used for video indexing, providing the user with efficient and intuitive links for accessing actions in videos. This is possible since every object includes the time of its appearance in the original video.

For indexing applications, the original video frames of active periods should be stored together with the object-based queue. Once a video synopsis is generated and an object from the synopsis is selected, the time associated with this object takes the viewer directly to the desired location in the original video.

An object can be selected by clicking on the area where it appears in the video synopsis with a mouse. For ease of object selection, playback options of pause and slow forward/backward can bring the synopsis video to the desired temporal position. The selected synopsis frame can be divided into regions, each region relating to a single active object in this frame. Selecting a region in the displayed synopsis frame will index into the desired time of the original video where the selected object appears.

5.2 Customized Energy Functions

In most cases, not all objects are of interest. A traffic surveillance camera may be interested only in cars, while other applications may prefer pedestrians. Filtering of objects can be done in several places. Objects can be filtered out before entering into the queue and, in this case, it will never be possible to retrieve them. Alternatively, objects can be filtered only at the query stage. In this case, the queue will include all objects and different queries can extract different objects from the queue. It is also possible to create a customized energy function for each application.

A simple example of customization is shown in Fig. 18b, where only small, dark objects were selected from the queue. While the original synopsis includes mostly cars, the new synopsis includes mostly pedestrians. Another example appears in Fig. 19, where the energy function included the element of a "phase transition" (Section 5.5), when a moving object stops and becomes part of the background.

5.3 Synopsis Specification

There are a few different possibilities for letting the users specify the features of a requested video synopsis:

1. Users can specify the desired duration of the video synopsis and the penalty for object collisions. In this case, the optimization stage will maximize the number of objects that will be included in the synopsis under the specified constraints.
2. Users can specify the desired duration of the video synopsis and the percentage of objects that must be included in the synopsis. The optimization stage will generate a video synopsis having minimum collisions under the specified constraints.
3. Users can specify the percentage of objects that must be included in the synopsis and the penalty for object collision. The optimization stage will minimize the duration of the synopsis under the specified constraints.

We have implemented option 1, where the duration of the video synopsis was determined by the user as a hard constraint. Surveillance video may prefer option 2 or 3, requiring that most objects will be included in the synopsis.

5.4 Object-Based Fast-Forward

Fast-forward is the most common tool used for video summarization and is always applied to entire frames. For example, "time-lapse" videos display, in a short time, slow processes like the growth of flowers, etc. Some current methods suggest an adaptive fast-forward [17], [23], but are still limited to the framework of entire frames. With video synopsis, each object can have its own "fast-forward" based on its importance or based on its original velocity. Slow objects may be accelerated, but not fast objects.

Object fast-forward can be done in a simple manner, e.g., bringing all moving objects to a uniform velocity. Alternatively, the speedup of slow objects can be determined during the optimization stage, giving some penalty to a speedup of objects. Adding object-based fast-forward to the optimization stage can further improve the temporal compression rate of the synopsis video at the expense of increasing the complexity of the optimization.

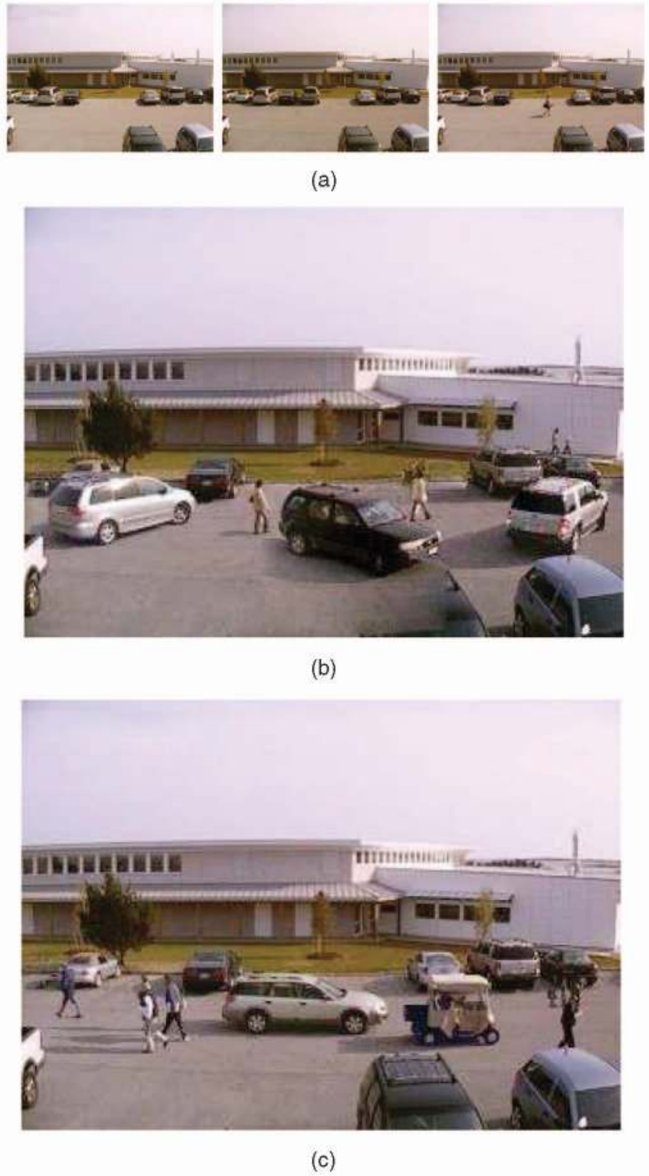

Fig. 19. (a) Three frames taken over 5 hours from a webcam watching a quiet parking lot. (b) A frame from a short synopsis of this period. A high score was given to phase transitions (e.g., moving objects that stop and become background). The video synopsis includes mostly cars involved in parking. (c) Objects without phase transitions are preferred. Only passing cars and pedestrians are shown in this synopsis.

5.5 Foreground-Background Phase Transitions

Phase transitions occur when a moving object becomes stationary and merges with the background or when a stationary object starts moving. Examples are cars being parked or getting out of parking. In most cases, phase transitions are significant events and we detect and mark each phase transition for use in the query stage.

We can find phase transitions by looking for background changes that correspond to the beginning and ending of tubes. These transitions are important as they explain the changes in the background. Fig. 19b shows a synopsis where objects that correspond to phase transitions are preferred. Mostly cars involved in parking are shown. In contrast, in Fig. 19c, objects that do not correspond to phase transitions are preferred. Only passing cars and pedestrians are shown.

Since phase transitions correspond to changes in the background, the stitching of phase transitions into the background should be given special attention. Two effects may occur in the synopsis video when phase transitions are not inserted into the background at the right time. 1) Background objects will appear and disappear with no reason, causing a flickering effect. 2) Moving objects will disappear when they stop moving rather than becoming part of the background. To minimize such effects in the video synopsis, phase transitions should be inserted into the time-lapse background at a time which corresponds to their original time.

6 Concluding Remarks

Video synopsis has been proposed as an approach for condensing the activity in a video into a very short time period. This condensed representation can enable efficient access to activities in video sequences and enable effective indexing into the video.

Two approaches were presented for video synopsis: One approach uses low-level graph optimization, where each pixel in the synopsis video is a node in this graph. This approach has the benefit of obtaining the synopsis video directly from the input video, but the complexity of the solution may be very high. An object-based approach detects and segments moving objects and performs the optimization on the detected objects. The object-based approach is much faster and enables the use of object-based constraints.

The activity in the resulting video synopsis is much more condensed than the activity in any ordinary video and viewing such a synopsis may seem awkward to the nonexperienced viewer. But, when the goal is to observe much information in a short time, video synopsis delivers this goal.

6.1 Computational Costs

Creating a video synopsis of an endless video stream has two major phases as shown in Fig. 13: an online phase and a query phase.

The online phase runs in parallel to video capture and recording and is independent of any query. In this phase, moving objects are detected and tracked and are entered as metadata into the object queue. Only frames with detected changes, caused by motion or by illumination, are further processed for extracting moving objects. The complexity of our object extraction is governed by the min-cut process and runs at 10 fps (on a 3 GHz PC) for frames of size 320 × 240. Since most surveillance videos include many frames with no activity that are automatically skipped, our implementation of this phase requires less than an hour to process a 1 hour video. Alternatively, hardware solutions for the detection and tracking of moving objects are provided by most surveillance companies ("VMD") and can be used instead. Since the first phase happens in parallel to video capture, it does not delay the response to a user query.

The response phase starts after a user presents a query to the system, specifying the Period of Interest (POI) in the input video and the length of the synopsis video. In this phase, all objects in the POI are selected and packed into the synopsis range by optimizing the target cost. This includes computing the cost function (20) and determining the temporal rearrangement of objects to minimize this cost. The most expensive element in the cost function is the collision cost (14), which is computed for every relative time shift between each pair of objects. Given K objects and T time steps, a naive computation of the collision cost includes $T * K^2$ computations of correlation between objects. Longer POI results in more objects (a larger K) and a longer synopsis video results in a larger T. The computation complexity can be significantly reduced by 1) using coarser time intervals (e.g., every 10 frames), 2) using reduced image resolution, and 3) using bounding boxes for each object in each frame to avoid the computation for pairs of objects (and time shifts) with no overlap. Cost computation took 65 seconds on the 334,000 frames of the parking scene (24 hours), having 262 objects, for a synopsis of length 450 frames. In the airport scene, with 100,000 frames covering 30 hours, the cost function for 500 objects was computed in 80 seconds.

Given the computed elements of the cost function, the optimal temporal arrangement is computed. Given K objects and T time steps, there are T^K possible arrangements. Greedy optimization converged to good results in the Parking example after 59 seconds and, for the Airport example, after 290 seconds (4.8 minutes).

The second phase is accelerated by removing in advance from the object queue objects that have a very small likelihood of being selected for a synopsis. For example, older or smaller objects may be considered as less interesting. Such objects can be removed as long as other objects with higher interest are available (Section 4.2). This stage, for example, decreased the number of objects from 1,917 to 500 in the Airport scene.

6.2 Limitations and Failures

Video synopsis is less applicable in several cases, some of which are listed below:

1. *Video with already dense activity.* All locations are active all the time. An example is a camera in a busy train station.
2. *Edited video, like a feature movie.* The intentions of the movie creator may be destroyed by changing the chronological order of events.

The object-based approach depends on object segmentation and tracking. While this task is relatively easy in the case of a static camera or even a rotating camera, it may be more difficult in the case of a moving camera constantly changing its viewing direction.

In some cases, the video synopsis is very condensed with objects and events, making it difficult for a user to search for any particular object. Making video synopsis that is easier to view is a topic for future studies.

Acknowledgments

This research was supported by the Israel Science Foundation, by the Israeli Ministry of Science, and by Google.

References

[1] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen, "Interactive Digital Photomontage," *Proc. ACM SIGGRAPH '04,* pp. 294-302, 2004.

[2] A. Agarwala, K.C. Zheng, C. Pal, M. Agrawala, M. Cohen, B. Curless, D. Salesin, and R. Szeliski, "Panoramic Video Textures," *Proc. ACM SIGGRAPH '05,* pp. 821-827, 2005.

[3] A. Agarwala, "Efficient Gradient-Domain Compositing Using Quadtrees," *Proc. ACM SIGGRAPH '07,* 2007.

[4] J. Assa, Y. Caspi, and D. Cohen-Or, "Action Synopsis: Pose Selection and Illustration," *Proc. ACM SIGGRAPH '05,* pp. 667-676, 2005.

[5] O. Boiman and M. Irani, "Detecting Irregularities in Images and in Video," *Proc. Int'l Conf. Computer Vision,* pp. I: 462-I: 469, 2005.

[6] Y. Boykov, V. Kolmogorov, "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 9, pp. 1124-1137, Sept. 2004.

[7] G. Brostow and I. Essa, "Motion Based Decompositing of Video," *Proc. Int'l Conf. Computer Vision,* pp. 8-13, 1999.

[8] S. Cohen, "Background Estimation as a Labeling Problem," *Proc. Int'l Conf. Computer Vision,* pp. 1034-1041, 2005.

[9] G. Doretto, A. Chiuso, Y. Wu, and S. Soatto, "Dynamic Textures," *Int'l J. Computer Vision,* vol. 51, pp. 91-109, 2003.

[10] A.M. Ferman and A.M. Tekalp, "Multiscale Content Extraction and Representation for Video Indexing," *Proc. SPIE,* vol. 3229, pp. 23-31, 1997.

[11] M. Irani, P. Anandan, J. Bergen, R. Kumar, and S. Hsu, "Efficient Representations of Video Sequences and Their Applications," *Signal Processing: Image Comm.,* vol. 8, no. 4, pp. 327-351, 1996.

[12] H. Kang, Y. Matsushita, X. Tang, and X. Chen, "Space-Time Video Montage," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1331-1338, June 2006.

[13] C. Kim and J. Hwang, "An Integrated Scheme for Object-Based Video Abstraction," *ACM Multimedia,* pp. 303-311, 2000.

[14] V. Kolmogorov and R. Zabih, "What Energy Functions Can Be Minimized via Graph Cuts?" *Proc. Seventh European Conf. Computer Vision,* pp. 65-81, 2002.

[15] Y. Li, T. Zhang, and D. Tretter, "An Overview of Video Abstraction Techniques," Technical Report HPL-2001-191, HP Laboratory, 2001.

[16] Y.-F. Ma, X.-S. Hua, L. Lu, and H. Zhang, "A Generic Framework of User Attention Model and Its Application in Video Summarization," *IEEE Trans. Multimedia,* vol. 7, no. 5, pp. 907-919, 2005.

[17] J. Nam and A. Tewfik, "Video Abstract of Video," *Proc. IEEE Third Workshop Multimedia Signal Processing,* pp. 117-122, Sept. 1999.

[18] J. Oh, Q. Wen, J. Lee, and S. Hwang, "Video Abstraction," *Video Data Management and Information Retrieval,* S. Deb, ed., pp. 321-346, Idea Group Inc. and IRM Press, 2004.

[19] M. Oren, C. Papageorgiou, P. Shinha, E. Osuna, and T. Poggio, "A Trainable System for People Detection," *Proc. Image Understanding Workshop,* pp. 207-214, 1997.

[20] M. Gangnet, P. Perez, and A. Blake, "Poisson Image Editing," *Proc. ACM SIGGRAPH '03,* pp. 313-318, July 2003.

[21] C. Pal and N. Jojic, "Interactive Montages of Sprites for Indexing and Summarizing Security Video," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* p. II: 1192, 2005.

[22] R. Patil, P. Rybski, T. Kanade, and M. Veloso, "People Detection and Tracking in High Resolution Panoramic Video Mosaic," *Proc. Int'l Conf. Intelligent Robots and Systems,* vol. 1, pp. 1323-1328, Oct. 2004.

[23] N. Petrovic, N. Jojic, and T. Huang, "Adaptive Video Fast Forward," *Multimedia Tools and Applications,* vol. 26, no. 3, pp. 327-344, Aug. 2005.

[24] A. Pope, R. Kumar, H. Sawhney, and C. Wan, "Video Abstraction: Summarizing Video Content for Retrieval and Visualization," *Signals, Systems, and Computers,* pp. 915-919, 1998.

[25] Y. Pritch, A. Rav-Acha, A. Gutman, and S. Peleg, "Webcam Synopsis: Peeking Around the World," *Proc. Int'l Conf. Computer Vision,* Oct. 2007.

[26] A. Rav-Acha, Y. Pritch, and S. Peleg, "Making a Long Video Short: Dynamic Video Synopsis," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 435-441, June 2006.

[27] A. Rav-Acha, Y. Pritch, D. Lischinski, and S. Peleg, "Dynamosaics: Video Mosaics with Non-Chronological Time," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 10, pp. 1789-1801, Oct. 2007.

[28] C. Rother, L. Bordeaux, Y. Hamadi, and A. Blake, "Autocollage," *ACM Trans. Graphics,* vol. 25, no. 3, pp. 847-852, July 2006.

[29] A. Schödl, R. Szeliski, D.H. Salesin, and I. Essa, "Video Textures," *Proc. ACM SIGGRAPH '00,* K. Akeley, ed., pp. 489-498, 2000.

[30] A.M. Smith and T. Kanade, "Video Skimming and Characterization through the Combination of Image and Language Understanding," *Proc. Int'l Workshop Content-Based Access of Image and Video Databases,* pp. 61-70, 1998.

[31] A. Stefanidis, P. Partsinevelos, P. Agouris, and P. Doucette, "Summarizing Video Datasets in the Spatiotemporal Domain," *Proc. 11th Int'l Workshop Database and Expert Systems Applications,* pp. 906-912, 2000.

[32] J. Sun, W. Zhang, X. Tang, and H. Shum, "Background Cut," *Proc. Ninth European Conf. Computer Vision,* pp. 628-641, 2006.

[33] M. Yeung and B.-L. Yeo, "Video Visualization for Compact Presentation and Fast Browsing of Pictorial Content," *IEEE Trans. Circuits and Systems for Video Technology,* vol. 7, no. 5, pp. 771-785, 1997.

[34] H. Zhong, J. Shi, and M. Visontai, "Detecting Unusual Activity in Video," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 819-826, 2004.

[35] X. Zhu, X. Wu, J. Fan, A.K. Elmagarmid, and W.G. Aref, "Exploring Video Content Structure for Hierarchical Summarization," *Multimedia Systems,* vol. 10, no. 2, pp. 98-115, 2004.

Yael Pritch received the BSc and MSc degrees in computer science from the Hebrew University of Jerusalem in 1998 and 2000, respectively, where she is currently a PhD student in computer science. Her research interests include computer vision with emphasis on motion analysis, video summary, and video manipulations.

Alex Rav-Acha received the BSc, MSc, and PhD degrees in computer science from the Hebrew University of Jerusalem in 1997, 2001, and 2008, respectively. He is currently a postdoctoral researcher in the Department of Computer Science and Applied Mathematics at the Weizmann Institute of Science, Rehovot, Israel. His research interests include video summary, video editing, image matting, and motion analysis.

Shmuel Peleg received the BSc degree in mathematics from the Hebrew University of Jerusalem in 1976 and the MSc and PhD degrees in computer science from the University of Maryland, College Park, in 1978 and 1979, respectively. He has been a faculty member at the Hebrew University of Jerusalem since 1980. His research interests include computer vision. He is a member of the IEEE.