Webcam Synopsis: Peeking Around the World*

Yael Pritch Alex Rav-Acha Avital Gutman Shmuel Peleg
School of Computer Science and Engineering
The Hebrew University of Jerusalem
91904 Jerusalem, Israel

Abstract

The world is covered with millions of webcams, many transmit everything in their field of view over the Internet 24 hours a day. A web search finds public webcams in airports, intersections, classrooms, parks, shops, ski resorts, and more. Even more private surveillance cameras cover many private and public facilities. Webcams are an endless resource, but most of the video broadcast will be of little interest due to lack of activity.

We propose to generate a short video that will be a synopsis of an endless video streams, generated by webcams or surveillance cameras. We would like to address queries like "I would like to watch in one minute the highlights of this camera broadcast during the past day". The process includes two major phases: (i) An online conversion of the video stream into a database of objects and activities (rather than frames). (ii) A response phase, generating the video synopsis as a response to the user's query.

To include maximum information in a short synopsis we simultaneously show activities that may have happened at different times. The synopsis video can also be used as an index into the original video stream.

1. Introduction

Millions of webcams and surveillance cameras are covering the world, capturing their field of view 24 hours a day. It is reported that in the UK alone there are 4.2 million security cameras covering city streets. Many webcams even transmit their video publicly over the Internet for everyone to watch. Several web sites try to index webcams by location or by functionality, and there is still much to be done in order to better organize this endless resource.

One of the problems in utilizing webcams is that they provide unedited raw data. A two hours 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, a viewer in one continent is likely to reach a webcam in another continent during hours of non-activity because of time-zone differences.

Our work tries 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. To enable this, 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 webcam the interesting objects, their duration, location, and their appearance. In a 3D space-time description of the video, each object is a "tube". In this paper we assume that moving objects are interesting, as well as phase transitions when a moving object turns into background and vice versa. Other criteria, e.g. using object recognition, can also be used to define objects of interest more accurately.

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 webcam broadcast during the last hour", or "I would like to watch in five minutes a synopsis of 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. To include as many activities as possible in the short video synopsis, objects may be displayed concurrently, even if they originally occurred at different times. The synopsis video can serve as an index into the original video by keeping a pointer to the original time for each object.

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.

1.1. Related Work

A video clip describes visual activities along time, and compressing the time axis allows viewing a summary of

*This research was supported (in part) by grants from the Israeli Ministry of Science and from Google.

978-1-4244-1631-8/07/$25.00 ©2007 IEEE

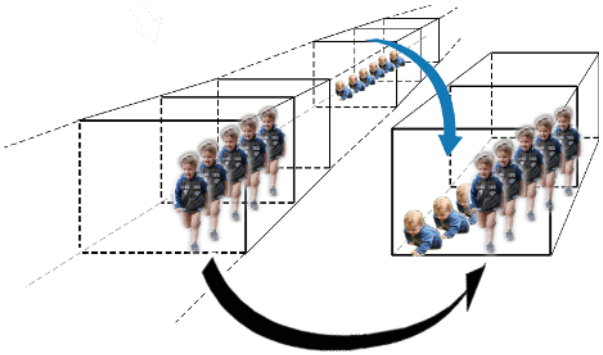

Figure 1. The input video stream has a baby and a child at two different times. A video synopsis can show the two objects simultaneously in a shorter video. We refer to the space-time location of an object as a *tube*.

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 [14, 20, 6]. 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 [24].

Many approaches to video summary eliminate completely the time axis, and show a synopsis of the video by selecting a few key frames [10, 28]. 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 [13, 15].

In both approaches above entire frames are used as the fundamental building blocks, and each frame is either shown completely or not shown at all. A different methodology uses mosaic images together with some meta-data for video indexing [8, 21, 18]. In this case the static synopsis image includes objects from different times.

Object-based approaches to video synopsis were first presented in [22, 9], where moving objects are represented in the space-time domain. Both papers introduce a new concept: creating a synopsis that combines objects that may have appeared at different times (See Fig 1). Our approach is most similar to the approach in [22], with the major difference that we address an infinite video stream rather than a short video clip.

In [23] infinite video is generated from a short video clip. While having a different goal, objects (video sprites) are separated from the background and rendered at arbitrary video locations to create novel videos.

1.2. Proposed Approach to Webcam Synopsis

A two phase process is proposed for webcam synopsis

1. **Online Phase** during video capture. This phase is done in real time.
   - Object (tube) detection in space time (Section 2).
   - Inserting detected tubes into the object queue (Section 4).
   - Removing tubes from the object queue when reaching a space limit (Section 4).
2. **Response Phase** building a synopsis according to a user query. This phase may take a few minutes, depending on the activity in the time period of interest.
   - Construction of time lapse video of the changing background (Section 5.1). Background changes can be caused, for example, by day-night differences.
   - Selection of tubes that will appear in the synopsis and their corresponding times (Section 3, Section 5.3).
   - Stitching the tubes and the background into a coherent video (Section 5.4). This step should take into account that activities from different times can appear simultaneously, and on a background from yet another time.

Since this work presents a video-to-video transformation, readers are encouraged to view the video examples in http://www.vision.huji.ac.il/webcam.

2. Computing Activity Tubes

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, 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 [16, 19]), dynamic textures [7], or detection of unusual activity [2, 27].

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

We used a simplification of [25] to compute the space-time tubes representing dynamic objects. The resulting tubes are connected components in the 3D space-time volume, and their generation is briefly described in the following subsections.

Figure 2. Four background images from a webcam at Stuttgart airport. The bottom images are at night, while the top images are at daylight. Notice that parked cars and parked airplanes become part of the background. This figure is best viewed in color.

Figure 3. Four extracted tubes shown "flattened" over the corresponding backgrounds from Fig. 2. 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.

2.1. Background Construction

The appearance of the background changes in time due to changes in lighting, changes of background objects, etc. To compute the background image for each time, we use a temporal median over a few minutes before and after each frame. We normally use a median over four minutes. Other methods for background construction are possible, even when using a shorter temporal window [5], but we used the median due to efficiency considerations.

Fig. 2 shows several background images as they vary during the day.

2.2. Moving Objects Extraction using Min-Cut

For extracting moving objects we follow [25] using background subtraction together with min-cut to get a smooth segmentation of foreground objects. In [25], image gradients that coincide with background gradients are attenuated, as they are less likely to be related to motion boundaries.

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 [3]:

$$E(f) = \sum_{r \in V} E_1(f_r) + \lambda \sum_{(r,s) \in N} E_2(f_r, f_s), \qquad (1)$$

where $E_1(f_r)$ is the color term, $E_2(f_r, f_s)$ is the contrast term between adjacent pixels r and s, and λ is a user defined weight. As a contrast term, we used the same term as [25] (Its description is omitted due to lack of space).

As for the 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:

$$\begin{aligned} E_1(1) &= \begin{cases} 0 & d_r > k_1 \\ k_1 - d_r & otherwise \end{cases}, \\ E_1(0) &= \begin{cases} \infty & d_r > k_2 \\ d_r - k_1 & k_2 > d_r > k_1 \\ 0 & otherwise \end{cases}, \end{aligned} \qquad (2)$$

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.

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

Each tube b is represented by its characteristic function

$$\chi_b(x,y,t) = \begin{cases} ||I(x,y,t) - B(x,y,t)|| & t \in t_b \\ 0 & otherwise \end{cases}, \qquad (3)$$

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.

2.3. Foreground-Background Phase Transitions

Tubes that abruptly begin or end in the middle of a frame represent phase transitions: A moving object that became stationary and has been merged with the background, or a stationary object that started 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 beginning and ending of tubes. Fig. 8 shows a synopsis where objects with frames transitions have higher preferences.

3. 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 between objects. The activity tubes are stored in the object queue B. Each tube b is defined over a finite time segment in the original video stream $t_b = [t_b^s, t_b^e]$.

The synopsis video is generated based on a temporal mapping M, shifting objects b in time from the original 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 $M(b) = \emptyset$. Optimal synopsis video will minimize the following energy function:

$$E(M) = \sum_{b \in B} E_a(\hat{b}) + \sum_{b,b' \in B} (\alpha E_t(\hat{b},\hat{b'}) + \beta E_c(\hat{b},\hat{b'})), \quad (4)$$

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 more dense video where objects may overlap. Increasing this weight will result in sparser video where objects do not overlap and less activity is presented.

3.1. 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

$$E_a(\hat{b}) = \sum_{x,y,t} \chi_{\hat{b}}(x,y,t), \quad (5)$$

where $\chi_b(x,y,t)$ is the characteristic function as defined in Eq. (3). For each tube b, whose mapping $\hat{b} = M(b)$ is partially included in the final synopsis, we define the activity cost similar to Eq. (5) but only pixels that were not entered into the synopsis are used.

3.2. 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:

$$E_c(\hat{b},\hat{b'}) = \sum_{x,y,t \in \hat{t_b} \cap \hat{t_{b'}}} \chi_{\hat{b}}(x,y,t)\chi_{\hat{b'}}(x,y,t) \quad (6)$$

Where $\hat{t_b} \cap \hat{t_{b'}}$ is the time intersection of b and b' in the synopsis video.

3.3. Temporal Consistency Cost

The temporal consistency cost adds a bias towards preserving the chronological order of events. The preservation of chronological order is more important for tubes that have a strong interaction. Therefore, the temporal consistency cost is weighted by the spatio-temporal interaction of each pair of tubes, $d(b,b')$, defined below.

$$\begin{array}{l} \text{if} \quad \hat{t_b} \cap \hat{t_{b'}} \neq \emptyset \quad \text{then} \\ d(b,b') = \exp(-\min_{t \in \hat{t_b} \cap \hat{t_{b'}}} \{d(b,b',t)\}/\sigma_{space}), \end{array} \quad (7)$$

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:

$$d(b,b') = \exp(-(\hat{t_{b'}^s} - \hat{t_b^e})/\sigma_{time}), \quad (8)$$

where σ_{time} is a parameter defining the extent of time in which events are still considered as having temporal interaction.

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

$$E_t(\hat{b},\hat{b'}) = d(b,b') \cdot \begin{cases} 0 & t_{b'}^s - t_b^s = \hat{t_{b'}^s} - \hat{t_b^s} \\ C & \text{otherwise} \end{cases}, \quad (9)$$

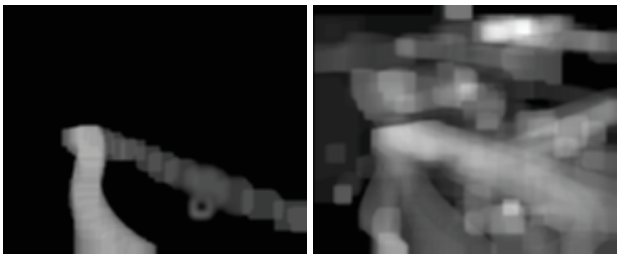

Figure 4. The activity distribution in the airport scene (intensity is log of activity value). The activity distribution of a single tube is on the left, and the average over all tubes is on the right. As expected, highest activity is on the auto lanes and the runway. Potential collision of tubes is higher in regions having a higher activity.

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

4. The Object Queue

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 (Eq. (5)) 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 according to three simple criteria that can be computed efficiently: "importance" (activity), "collision potential", and "age".

A possible measure for the importance of an object is the sum of its characteristic function as defined in Eq. (5).

Since the collision cost can not 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. 4.

A possible approach to address the removal of older objects is to 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 the t's interval, N_t, should be proportional to

$$N_t = K\frac{1}{\sigma}e^{-\frac{t}{\sigma}}, \tag{10}$$

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.

5. 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. (i) Generating a background video. (ii) Once the background video is defined, a consistency cost is computed for each object and for each possible time in the synopsis. (iii) An energy minimization step determines which tubes (space-time objects) appear in the synopsis and at what time. (iv) 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 user query a second (smaller) object queue is generated, having only objects from the desired time period. To enable fast optimization, the collision cost (Eq (6)) between every two objects in the smaller queue is computed.

5.1. Time Lapse Background

The basis for the synopsis video is a time lapse background video, generated before adding activity tubes into the synopsis. The background video has two tasks: (i) It should represent the background changes over time (e.g. day-night transitions, etc.). (ii) It should represent the background for 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. (i) A temporal activity histogram H_a of the video stream. (ii) 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.

5.2. 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-background consistency can be taken into account by adding a new energy term $E_b(M)$. This term will measure the cost of stitching this object to the time-lapsed 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 lapsed background. we set:

$$E_s(\hat{b}) = \sum_{x,y\in\sigma(\hat{b}),t\in\hat{t_b}\cap t_{out}} \|I_{\hat{b}}(x,y,t) - B_{out}(x,y,t)\|, \tag{11}$$

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).

5.3. Energy Minimization

To crate the final synopsis video we look for a temporal mapping M that minimizes the energy in Eq. (4) together with the background consistency term in Eq. (11), giving

$$\begin{aligned} E(M) \quad &= \textstyle\sum_{b\in B}(E_a(\hat{b}) + \gamma E_s(\hat{b}))+ \\ &+ \textstyle\sum_{b,b'\in B}(\alpha E_t(\hat{b},\hat{b'}) + \beta E_c(\hat{b},\hat{b'})), \end{aligned} \tag{12}$$

where α, β, γ are user selected weights that are query dependent. Since the global energy function (12) is written as a sum of energy terms defined on singles or pairs of tubes, it can be minimized by various MRF-based techniques such as [26, 12]. In our implementation we used the simpler simulated annealing method [11] which gave good results. The simulated annealing works 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 subset of tubes that are included in the synopsis, and 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.

In order to make the solution feasible, we restricted the temporal shifts of tubes to be in jumps of 10 frames.

5.4. Stitching the Synopsis Video

The stitching of tubes from different time periods poses a challenge to existing methods such as [1]. Stitching all the tubes at once results in a blending of colors from different backgrounds. Therefore, we take a slightly different approach: Each tube is stitched independently to the time lapse background. Any blending method is possible and in our experiments we used Poisson editing [17]. Overlapping tubes are blended together by letting each pixel be a weighted average of the corresponding pixels from the stitched blobs $\hat{b}$, 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 assumes that an object whose vertical image position is lower is also closer. Other depth ordering methods include [4]. The frequency of object occlusion cases depends on the relative weights of the collision cost (that prevent such cases) in respect to other costs.

6. Examples

We have applied the webcam synopsis to 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.

Fig. 5, and Fig. 7 are from cameras stationed outdoors, while Fig. 6 is from a camera stationed indoors with constant lighting. In all these examples the main "interest" in each tube has been the number of moving pixels in it.

Fig. 8 shows the use of phase transitions for "interest". Important tubes are those that terminate with the object joining the background, or tubes of background objects that started moving. In the street case, parking cars or cars pulling out of parking are more likely to be shown in the synopsis.

7. Concluding Remarks

A method to create a short video that is a synopsis of an endless video stream has been presented. The method includes two phases. In the input phase the video stream is analyzed and objects of interest are detected and segmented from their background. While we presented an object interest function that is based on motion, any other approach for object detection, recognition, and segmentation can be used for the generation of the "tubes" - the 3D space-time representation of each object. This phase is carried out in real time.

Figure 5. Top: Three images taken over two days from a webcam at the (quiet) Stuttgart airport Bottom: A frame from a 1 minute synopsis of this period.

Figure 6. Top: Three images taken over two days from a webcam in a "French Billiard" club (no pockets, one player) Bottom: A frame from a 1 minute synopsis of this period. Notice the multiple players per table at the synopsis.

Queue management is necessary to bridge the gap between an infinite video and a finite storage. Several methodologies were described for determining which objects should be removed from the queue once it becomes full. But other methodologies are possible, and even a random selection of objects for removal from the queue may work fine.

Figure 7. Top: Three images taken overnight from a webcam in St. Petersburg Bottom: A frame from a 1 minute synopsis of this period. While this street is very busy during the day, it is practically deserted during the night. The video synopsis brings together many cars that passed this street during different times.

Figure 8. Top: Three images taken over several hours from a webcam watching a very quiet street Bottom: A frame from a 1 minute synopsis of this period. In this synopsis we have given a high score to phase transitions (e.g. moving objects that stop and become background), so the video synopsis includes mostly cars being parked or pulling out of parking.

The Second phase occurs after the user's query is given. A subset of the queue is extracted based on the target period

of interest, and the object tubes are arranged (by temporal shifts) to generate the optimal video synopsis. This stage delivers the video synopsis to the user, and currently it take a few minutes to compute.

Some very interesting aspects concern periodicity in background. The most obvious periods that can be easily detected are the day-night periods. In most cases when a few days are covered by a single synopsis, the time-lapse background may cover only a single day, while the activities will come from all days. This should be an option given to the user specifying the query.

While the examples in this paper address only stationary cameras, video synopsis can also be used with moving cameras. As in prior work that addressed video summary for moving cameras, this can be done together with methods to compute camera motion and object tracking.

References

[1] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Interactive digital photomontage. In *SIGGRAPH*, pages 294–302, 2004.

[2] O. Boiman and M. Irani. Detecting irregularities in images and in video. In *ICCV*, pages I: 462–469, Beijing, 2005.

[3] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, Sep. 2004.

[4] G. Brostow and I. Essa. Motion based decompositing of video. In *ICCV'99*, pages 8–13, Corfu, 1999.

[5] S. Cohen. Background estimation as a labeling problem. In *ICCV'05*, pages 1034–1041, Washington, DC, 2005.

[6] A. Divakaran, K. Peker, R. Radhakrishnan, Z. Xiong, and R. Cabasson. Video summarization using mpeg-7 motion activity and audio descriptors. Technical Report TR-2003-34, MERL - A Mitsubishi Electric Research Laboratory, Cambridge, Massachusetts, May 2003.

[7] G. Doretto, A. Chiuso, Y. Wu, and S. Soatto. Dynamic textures. *Int. J. Computer Vision*, 51:91–109, 2003.

[8] M. Irani, P. Anandan, J. Bergen, R. Kumar, and S. Hsu. Efficient representations of video sequences and their applications. *Signal Processing: Image Communication*, 8(4):327–351, 1996.

[9] H. Kang, Y. Matsushita, X. Tang, and X. Chen. Space-time video montage. In *CVPR'06*, pages 1331–1338, New-York, June 2006.

[10] C. Kim and J. Hwang. An integrated scheme for object-based video abstraction. In *ACM Multimedia*, pages 303–311, New York, 2000.

[11] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 4598(13):671–680, 1983.

[12] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? In *ECCV*, pages 65–81, 2002.

[13] Y. Li, T. Zhang, and D. Tretter. An overview of video abstraction techniques. Technical Report HPL-2001-191, HP Laboratory, 2001.

[14] J. Nam and A. Tewfik. Video abstract of video. In *3rd IEEE Workshop on Multimedia Signal Processing*, pages 117–122, Copenhagen, Sept. 1999.

[15] J. Oh, Q. Wen, J. lee, and S. Hwang. Video abstraction. In S. Deb, editor, *Video Data Mangement and Information Retrieval*, pages 321–346. Idea Group Inc. and IRM Press, 2004.

[16] M. Oren, C. Papageorgiou, P. Shinha, E. Osuna, , and T. Poggio. A trainable system for people detection. In *Proceedings of Image Understanding Workshop*, pages 207–214, 1997.

[17] M. G. P. Perez and A. Blake. Poisson image editing. In *SIGGRAPH*, pages 313–318, July 2003.

[18] C. Pal and N. Jojic. Interactive montages of sprites for indexing and summarizing security video. In *Video Proceedings of CVPR05*, page II: 1192, 2005.

[19] R. Patil, P. Rybski, T. Kanade, and M. Veloso. People detection and tracking in high resolution panoramic video mosaic. In *Int. Conf. on Intelligent Robots and Systems (IROS 2004)*, volume 1, pages 1323–1328, October 2004.

[20] N. Petrovic, N. Jojic, and T. Huang. Adaptive video fast forward. *Multimedia Tools and Applications*, 26(3):327–344, August 2005.

[21] A. Pope, R. Kumar, H. Sawhney, and C. Wan. Video abstraction: Summarizing video content for retrieval and visualization. In *Signals, Systems and Computers*, pages 915–919, 1998.

[22] A. Rav-Acha, Y. Pritch, and S. Peleg. Making a long video short: Dynamic video synopsis. In *CVPR'06*, pages 435–441, New-York, June 2006.

[23] A. Schödl, R. Szeliski, D. H. Salesin, and I. Essa. Video textures. In K. Akeley, editor, *Siggraph 2000, Computer Graphics Proceedings*, pages 489–498. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.

[24] A. M. Smith and T. Kanade. Video skimming and characterization through the combination of image and language understanding. In *CAIVD*, pages 61–70, 1998.

[25] J. Sun, W. Zhang, X. Tang, and H. Shum. Background cut. In *ECCV*, pages 628–641, 2006.

[26] Y. Weiss and W. Freeman. On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs. *IEEE Transactions on Information Theory*, 47(2):723–735, 2001.

[27] H. Zhong, J. Shi, and M. Visontai. Detecting unusual activity in video. In *CVPR*, pages 819–826, 2004.

[28] X. Zhu, X. Wu, J. Fan, A. K. Elmagarmid, and W. G. Aref. Exploring video content structure for hierarchical summarization. *Multimedia Syst.*, 10(2):98–115, 2004.