

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
{yaelpri, alexis, gutmant, peleg}@cs.huji.ac.il

Abstract

The world is covered with millions of webcams. Some are private, but 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.

These public webcams are an endless resource, and some sites are already mapping them by location or by functionality. But when a webcam is selected - most of the video broadcast will be of no interest due to lack of activity.

We propose to generate a short video that will be a synopsis of an infinite video stream, such as generated by a webcam. 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 two major phases are: (i) An online conversion of the video stream into a searchable structure based on 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, restoring the chronological order.

1. Introduction

Millions of webcams 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. Public webcams are available in universities, museums, zoos, ski resorts, traffic webcams, etc. Many security cameras are also online in stores, airports, and other public areas. Webcams are even transmitting video streams from private homes and

offices. 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 raw, unedited, data. A two hours movie, 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. For example, a webcam transmitting from one of Nasa’s control rooms shows no activity most of the time, but is very interesting at times of space missions. Also, a viewer in one continent is likely to reach a webcam in another continent during hours of non-activity because of time differences.

Our work tries to make the webcam resource more useful by giving the viewer the ability to view summaries of the infinite 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 space-time volume. In this work we assume that moving objects are interesting, as well as phase transitions when a moving object turn into background and vice versa. Other criteria, e.g. involving object recognition, can always be used to define objects of interest.

A query that could be answered by the system can 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. A pointer to the original time is available for each object, and a video with the original chronological timing can be

*This research was supported by the Israel Science Foundation, Grant number 354/04, and by Google.

generated from the stored objects.

While webcam video is endless, and the number of objects is not bound, storage available for each webcam may be limited. We address the issue of keeping a finite queue of objects, and propose a procedure for removing objects from this queue when space is needed for new objects.

1.1. Related Work

A video clip describes visual activities along time, and compressing the time axis allows viewing 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 summary. A special case of fast-forward is called “time lapse”, which generates 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 [12, 18, 4]. 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 [21].

Many approaches to video summary eliminate completely the time axis, and show a synopsis of the video by selecting a few key frames [8, 24]. 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 [11, 13].

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 [6, 19, 16]. In this case the static synopsis image includes objects from different times.

Object-based approaches to video synopsis were presented in [20, 7], where moving objects were represented in space-time domain. Both papers introduced 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 [20], with the major difference that we address an infinite video stream rather than a short video clip. This and other differences will be described in the next section.

1.2. Unique Challenges in Webcam Synopsis

The observation that more activities can be shown in a shorter video, if the chronological order is not enforced, was first made in [20, 7]. They took a short video clip and made it shorter. While we also relax the chronological order as was done in these papers, addressing an infinite video involves many additional challenges:

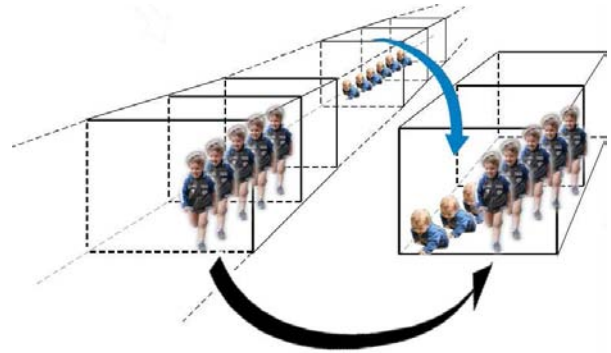


Figure 1. The input video stream has a baby and a child at two different times. A video synopsis can show the two objects simultaneously in a shorter video. We refer to the space-time location of an object as a *tube*.

- Since no storage is infinite, there is a need to “forget” events when an infinite video is summarized.
- The appearance of the background varies substantially in a long video, e.g. day to night. These changes should be addressed when creating the background of the synopsis, and when inserting objects into the background.
- Because activities from different times can appear simultaneously, and on a background from even another time, special care should be made when stitching all these to give the output video.
- The selection of the tubes to be shown and their placement in the synopsis video must be take into account many variables and more objects.
- Fast response to user queries in spite of the huge amount of data.

Since this work presents a video-to-video transformation, the reader is encouraged to view the video examples in www.vision.huji.ac.il/webcam/.

2. Computing Activity Tubes

In order to generate a useful synopsis, we need to identify the space-time location of interesting objects and activities (*tubes*). In most 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 easy to incorporate object recognition (e.g. people detection [14, 17]) and dynamic textures [5].



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.

Space-time tubes representing dynamic objects are computed using background subtraction followed by object segmentation (we used min-cut). The resulting tubes are connected components in the space-time volume, and their generation is described in the following subsections.

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, but this can change for each webcam depending on the speed of objects in it. Other methods for background construction are possible, even when using a shorter temporal window [3], but we found the median to be the fastest.

Fig. 2 shows several background images as they change during different times of the day.

2.2. Moving Objects Extraction using Min-Cut

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 labelling 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 [2]:

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

where $E_1(f_r)$ is the color term, $E_2(f_r, f_s)$ is the contrast



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.

term between adjacent pixels r and s , and λ is a user defined weight. Let $d_r = \|I(r) - B(r)\|$ be the color differences between the image I and the current background B . The foreground energy of a pixel r is set to

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

where k_1 and k_2 are thresholds (we used $k_1 = 30/255$ and $k_2 = 60/255$). The background energy of r is set to

$$E_1(1) = \begin{cases} 0 & d_r > k_1 \\ k_1 - d_r & \text{otherwise} \end{cases}. \quad (3)$$

Unlike other methods, we do not use a lower threshold since our process is robust to pixels that were wrongly identified as foreground, but we wish to prevent pixels wrongly identified as background.

A contrast term that encourages cutting along color gradients is

$$E_2(f_r, f_s) = \exp(-d_{rs}/2\beta) \quad (4)$$

where $d_{rs} = \|I_r - I_s\|^2$ is the color difference between adjacent pixels r and s , and β is the average of $\|I_r - I_s\|^2$ over the image. We actually use d_{rs} as described in [22], which attenuates gradients that appear also in the background.

After computing in all frames the masks of the moving objects using min-cut, we apply morphological dilation on the masks. All the masks in the 3D space time volume are grouped to connected components, denoted as “activity tubes”; Examples of extracted tubes are shown in Fig. 3.

Each tube b is defined by its characteristic function $\chi_b(x, y, t)$, which is zero at background pixels and equals

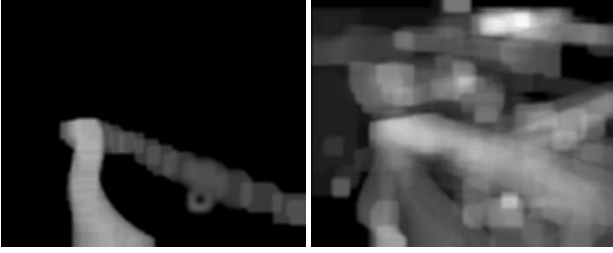


Figure 4. The activity distribution in the airport scene (intensity is log of activity value). The activity distribution of a single tube is on the left, and the average over all tubes is on the right. As expected, highest activity is on the auto lanes and runway. Potential collision of tubes is higher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

to d_r , the intensity difference from the background, at locations and times when 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 therefore 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. 9 shows a synopsis where objects with frames transitions have higher preferences.

3. The Object Queue

All detected objects, represented as tubes in the space-time volume, are stored in a queue awaiting user queries. 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 from the queue, we prefer to remove objects that are least likely to be included in a synopsis according to the ‘‘Activity Cost’’ and the ‘‘Collision Cost’’ as defined in Sec. 4. We therefore remove the object whose removal results in the lowest energy.

Instead of computing the collision cost between each pair of objects, we give a simpler penalty for ‘‘potential collisions’’. We compute an image representing the spatial activity distribution in the scene. This image is the sum of active pixels of all objects in each spatial location, normalized to sum to one. We also compute a similar spatial activity distribution for each individual object (this time unnormalized). The correlation between the two activity distributions is the potential collision cost for this object. An image showing the ‘‘activity distribution’’ in a scene is shown in Fig. 4.

Let us assume that the collision and activity costs of objects are independent of time. In this case, the probability of an object to remain in the queue will be reduced exponentially with its age: The older objects will have to survive more possible deletions. We can also get any desired temporal distribution of remaining objects. This is done by comparing the current temporal distribution of objects with the desired temporal distribution. Only objects from times where current object distribution is higher than desired will be candidates for removal.

4. Energy Between Objects

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

We look for a temporal mapping M , placing objects b from the original video into the time segment $\hat{t}_b = [\hat{t}_b^s, \hat{t}_b^e]$ in the video synopsis. $M(b) = \hat{b}$ indicates the mapping of tube b into the synopsis, and when b is not mapped to the output synopsis $M(b) = \emptyset$. We wish to minimize the following energy function:

$$E(M) = \sum_{b \in B} E_a(\hat{b}) + \sum_{b, b' \in B} (\alpha E_t(\hat{b}, \hat{b}') + \beta E_c(\hat{b}, \hat{b}')), \quad (5)$$

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.

4.1. Activity Cost

The activity cost favours synopsis movies with maximum activity. A penalty is given to synopsis movies for objects that are not mapped to a valid time in the synopsis. When a tube is excluded from the synopsis, $M(b) = \emptyset$, then

$$E_a(\hat{b}) = \sum_{x, y, t} \chi_{\hat{b}}(x, y, t). \quad (6)$$

For each tube b , whose mapping $\hat{b} = M(b)$ is partially included in the final synopsis, we define the activity cost as the sum of its pixels that were not entered into the synopsis:

$$E_a(\hat{b}) = \sum_{x, y, (t_b \setminus t_{out})} \chi_{\hat{b}}(x, y, t), \quad (7)$$

Where t_{out} is the duration of the output synopsis.

4.2. Collision Cost

For every two 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:

$$E_c(\hat{b}, \hat{b}') = \sum_{x,y,t \in \hat{t}_b \cap \hat{t}_{b'}} \chi_{\hat{b}}(x,y,t) \chi_{\hat{b}'}(x,y,t) \quad (8)$$

Where $\hat{t}_b \cap \hat{t}_{b'}$ is the time intersection of b and b' in the synopsis video.

As noted before, all tubes include some background pixels due to our morphological dilation of the masks (computed by the min-cut). Such pixels have smaller weights according to $\chi_{\hat{b}}$.

4.3. 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 which 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.

$$\text{if } \hat{t}_b \cap \hat{t}_{b'} \neq \emptyset \text{ then} \\ d(b, b') = \exp(-\min_{t \in \hat{t}_b \cap \hat{t}_{b'}} \{d(b, b', t)\} / \sigma_{space}), \quad (9)$$

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:

$$d(b, b') = \exp(-(t_{b'}^s - t_b^e) / \sigma_{time}), \quad (10)$$

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 objects to be played in the synopsis at the same order as in the original video. It adds a cost whenever the temporal order has been violated.

$$E_t(\hat{b}, \hat{b}') = d(b, b') \cdot \begin{cases} 0 & t_{b'}^s - t_b^e = t_{b'}^s - t_b^s \\ C & \text{otherwise} \end{cases}, \quad (11)$$

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

5. Synopsis Generation

The queue of tubes 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) The generation of background video. (ii) Once the background video is defined, each object in the queue gets a consistency cost to 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 for the final synopsis. These steps are described in this section.

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. This background video should represent the background changes over time (day-night transitions, etc.) and should also allow insertions of activity tubes. These goals are conflicting, as for better insertion we should spend more time showing 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:

$$E_s(\hat{b}) = \sum_{x,y \in \sigma(\hat{b}), t \in \hat{t}_b \cap t_{out}} \|I_{\hat{b}}(x, y, t) - B_{out}(x, y, t)\|, \quad (12)$$

where $\sigma(\hat{b})$ is the set of pixels in the border of the mapped activity tube \hat{b} . 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 create the final synopsis video we look for a temporal mapping M that minimizes the energy in Eq. (5) together with the background consistency term in Eq. (12), giving

$$E(M) = \sum_{b \in B} (E_a(\hat{b}) + \gamma E_s(\hat{b})) + \sum_{b, b' \in B} (\alpha E_t(\hat{b}, \hat{b}') + \beta E_c(\hat{b}, \hat{b}')), \quad (13)$$

where α, β, γ are user selected weights that are query dependent. Since the global energy function (13) 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 [23, 10]. In our implementation we used the simpler simulated annealing method [9] 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. Additional acceleration was obtained by using a second queue, similar to the one described in Section 3, to reduce the number of objects in the optimization stage. As in the large queue, this small queue uses a fast approximated score to remove objects that are not likely to be included in a synopsis. Since the query is already known at this stage, this queue differs from the first queue in two main aspects: (i) only objects that occur during the desired time period are considered. (ii) The time distribution of tubes over the desired time period is uniform.

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



Figure 5. Top: Three images taken over two days from a webcam in Cambridge University botanical gardens www.opentopia.com/showcam.php?camid=2184. Bottom: A frame from a 1 minute synopsis of this period.

lapse background using any blending method. In our experiments we used Poisson editing [15]. Then, all the 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 $\chi_{\hat{b}}(x, y, t)$. Transparency can be avoided by taking the pixel with maximal activity measure instead of the weighted average.

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 neighbourhood we do not count the number of frames, but we use the absolute times of each frame.

Fig. 5, Fig. 6, and Fig. 8 are from cameras stationed outdoors, while Fig. 7 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. 9 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.

Since this work presents a video-to-video transformation, the reader is encouraged to view the video examples in www.vision.huji.ac.il/webcam/.



Figure 6. Top: Three images taken over two days from a webcam at the (quiet) Stuttgart airport www.opentopia.com/showcam.php?camid=283. Bottom: A frame from a 1 minute synopsis of this period.



Figure 7. Top: Three images taken over two days from a webcam in a “French Billiard” club (no pockets, one player) www.opentopia.com/showcam.php?camid=4546. Bottom: A frame from a 1 minute synopsis of this period. Notice the multiple players per table at the synopsis.

7. Concluding Remarks

A method to create a short video that is a synopsis of an infinite video stream has been presented. The method includes three phases. In the input phase the video stream is



Figure 8. Top: Three images taken overnight from a webcam in St. Petersburg www.opentopia.com/showcam.php?camid=4567. 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.

analyzed and objects of interest are detected and segmented from their background. While we presented object interest 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 can (and should) be carried out in real time.

The second phase deals with queue management, 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 may work fine.

The third 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 takes a few minutes to compute.

Some very interesting aspects concern periodicity in background. For example, a video of the changing background is generated for each video stream. 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



Figure 9. Top: Three images taken over several hours from a webcam watching a very quiet street www.opentopia.com/showcam.php?camid=3931. 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.

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. When the background changes due to the motion of the camera, objects may be placed in different locations than their original locations. In such cases the objects may be placed over synthetic backgrounds.

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] 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.
- [3] S. Cohen. Background estimation as a labeling problem. In *ICCV’05*, pages 1034–1041, Washington, DC, 2005.
- [4] 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.
- [5] G. Doretto, A. Chiuso, Y. Wu, and S. Soatto. Dynamic textures. *Int. J. Computer Vision*, 51:91–109, 2003.
- [6] 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.
- [7] H. Kang, Y. Matsushita, X. Tang, and X. Chen. Space-time video montage. In *CVPR’06*, pages 1331–1338, New-York, June 2006.
- [8] C. Kim and J. Hwang. An integrated scheme for object-based video abstraction. In *ACM Multimedia*, pages 303–311, New York, 2000.
- [9] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 4598(13):671–680, 1983.
- [10] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? In *ECCV*, pages 65–81, 2002.
- [11] Y. Li, T. Zhang, and D. Tretter. An overview of video abstraction techniques. Technical Report HPL-2001-191, HP Laboratory, 2001.
- [12] J. Nam and A. Tewfik. Video abstract of video. In *3rd IEEE Workshop on Multimedia Signal Processing*, pages 117–122, Copenhagen, Sept. 1999.
- [13] J. Oh, Q. Wen, J. Lee, and S. Hwang. Video abstraction. In S. Deb, editor, *Video Data Management and Information Retrieval*, pages 321–346. Idea Group Inc. and IRM Press, 2004.
- [14] 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.
- [15] M. G. P. Perez and A. Blake. Poisson image editing. In *SIGGRAPH*, pages 313–318, July 2003.
- [16] C. Pal and N. Jojic. Interactive montages of sprites for indexing and summarizing security video. In *Video Proceedings of CVPR05*, page II: 1192, 2005.
- [17] 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.
- [18] N. Petrovic, N. Jojic, and T. Huang. Adaptive video fast forward. *Multimedia Tools and Applications*, 26(3):327–344, August 2005.
- [19] 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.
- [20] 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.
- [21] A. M. Smith and T. Kanade. Video skimming and characterization through the combination of image and language understanding. In *CAIVD*, pages 61–70, 1998.
- [22] J. Sun, W. Zhang, X. Tang, and H. Shum. Background cut. In *ECCV*, pages 628–641, 2006.

- [23] 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.
- [24] 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.