VIDEO CONDENSATION BY RIBBON CARVING

Zhuang Li, Prakash Ishwar and Janusz Konrad

September 30, 2008

Boston University

Department of Electrical and Computer Engineering

Technical Report No. ECE-2008-03

BOSTON

UNIVERSITY

VIDEO CONDENSATION BY RIBBON CARVING

Zhuang Li, Prakash Ishwar and Janusz Konrad

Boston University
Department of Electrical and Computer Engineering
8 Saint Mary's Street
Boston, MA 02215
www.bu.edu/ece

September 30, 2008

Technical Report No. ECE-2008-03

This material is based upon work supported by the US National Science Foundation (NSF) under awards CNS–0721884 and (CAREER) CCF–0546598. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

Summary

Efficient browsing of long video sequences is a key tool in visual surveillance, e.g., for post-event video forensics, but can also be used for review of motion pictures and home videos. While frame skipping (fixed or adaptive) is straightforward to implement, its performance is quite limited. More efficient techniques have been developed, such as video summarization and video montage but they lose either the temporal or semantic context of events. A recently-proposed method called video synopsis provides even better performance, however, it involves multiple processing stages and is fairly complex. *Video condensation*, that we propose here, is novel in the way information is removed from the space-time video volume, is conceptually simple and relatively easy to implement. We introduce the concept of a *video ribbon* inspired by that of a seam recently proposed for image resizing. We recursively carve ribbons out by minimizing an activity-aware cost function using dynamic programming. The ribbon model we develop is flexible and permits an easy adjustment of the compromise between temporal condensation ratio and anachronism of events. We demonstrate ribbon carving efficiency on motor and pedestrian traffic videos.

Contents

List of Figures

List of Tables

1 Introduction

In the last decade, millions of cameras have been deployed in transportation hubs (e.g., airports, railway and bus stations), on city streets and highways, and in office buildings. In fact, in 2007 there were reported 30 million video cameras in use in the United States alone, producing over 4 billion hours of video footage each week [21]. London, UK is believed to be the most camera-saturated city in the world. This proliferation of surveillance cameras is largely due to the arrival of an inexpensive network camera, usually CMOS-based with dedicated video compression chip and wireline/wireless communication interface. Easily deployed and remotely managed, network cameras generate vast amounts of visual data, which makes continuous monitoring by human operators impractical due to high cost and reliability issues (e.g., operator fatigue). Therefore, of interest is the development of automatic algorithms to aid human operators, by identifying relevant data segments, and, eventually, to replace them [6, 9, 20].

This paper focuses on one class of such algorithms that compute a digest of video activities in the field of view of a surveillance camera. A video digest is an abbreviated video (for example, a 1-minute summary video produced from 1-hour original video) that preserves the most relevant activities while removing relatively static temporal and/or spatial segments. A video digest permits fast browsing through recorded surveillance video thus increasing efficiency in video forensics, but it can also be used for quick review of motion pictures or home videos [12].

Various approaches to computing video digests have been proposed to date with the most prominent being:

- fast forwarding, where individual frames or groups of frames are skipped in fixed intervals or adaptively [11, 13, 22],
- video summarization, where key frames are extracted and usually presented simultaneously as a storyline [12], and
- video montage, where relevant spatio-temporal (3-D) segments are extracted

and combined into a spatio-temporal "puzzle" [7, 8, 14–16].

While fast forwarding is conceptually simple, it has limited capabilities since only complete frames can be removed. As a consequence, the *video condensation ratio*, defined as the ratio of video length prior to processing to that after processing, is relatively low. Video summarization, by design, loses the dynamics of the original video as only key frames are presented. Video montage can be quite complex and may result in loss of context as objects are displaced both in time and in spatial coordinates. Although a variant, called video synopsis, has been proposed that permits only temporal shifts while creating the spatio-temporal "puzzle" [14], object displacements may cause a reversal of the order of activities. We discuss the state of the art in video abstraction in detail in the next section.

In this paper, we propose a novel approach to computing video digests that we call *video condensation*. Our approach, like that in video synopsis, permits only a temporal transformation of activities but, unlike in video synopsis, explicitly controls the temporal warping. Moreover, our approach does not require explicit extraction of objects and therefore is simpler to implement and efficient computationally. We make the following four contributions in this paper:

1. we introduce the concept of a *ribbon* in space-time video volume and apply it to activity-aware video condensation as a novel extension of the concept of a seam in 2-D images used for content-aware image resizing [2],

2. we introduce a ribbon flex-parameter to trade off the video condensation ratio and anachronism of events,

3. we propose an activity-based, instead of intensity-based, cost function for ribbon carving and an activity-adaptive stopping criterion for automatic control of the condensation ratio,

4. we develop a hierarchical, sliding-window approach to process endless video at multiple levels of condensation.

The paper is organized as follows. In Section 2, we review prior work on video digests. In Section 3, we summarize image seam carving that inspired this work. In Section 4, we introduce ribbons and ribbon carving, as well as their application to video condensation. Finally, in Section 5, we present experimental results, and in Section 6 we draw conclusions and list further challenges.

2 Prior work on video digests

The computation of a video digest can be as simple as skipping frames at regular intervals (e.g., every second frame for 2:1 condensation ratio). However, higher condensation ratios, of interest here, necessitate skipping numerous consecutive frames thus potentially leading to the exclusion of relevant activities. Although content-adaptive (irregular) skipping of frames has been developed [11, 13, 22], it removes only complete frames that are inactive or exhibit low activity, thus limiting the realizable condensation ratios.

Very high condensation ratios can be achieved by means of video summarization where key frames are extracted and presented sequentially, like a slide show, or simultaneously as a storyline. In the former case, temporal continuity of events is compromised due to, usually, large temporal gaps between consecutive key frames while in the latter case the time aspect is removed altogether. A related idea is that of video skimming where short representative sub-sequences from the original video are extracted and combined into a dynamic summary. This approach is a special case of content-adaptive skipping with the same deficiency of removing only complete frames which limits the attainable degree of condensation. A good review of video summarization and skimming can be found in [12].

The approaches discussed above remove/preserve complete video frames. An alternative is extracting spatial segments from different frames to combine them into a new entity. In one approach, spatial segments from different times are combined into a single mosaic image [7]. This results in a complete loss of dynamics but gives a

high-level view of the captured video. Alternatively, spatial segments from different times can be combined into a new video sequence by shifting the segments in time. One example is dynamic mosaicing where different-time video segments are aligned temporally to produce a wide-angle view of a dynamic scene (e.g., using a panning camera) [15]. In this case, the spatial location of segments is maintained thus enlarging the field of view captured. In another example, known as video montage, the spatial location of segments is modified in order to "pack" the output video with as many events as possible [8]. This, unfortunately, creates visible seams due to the combination of largely uncorrelated segments.

Recently, a variant of video montage, referred to as video synopsis, has been proposed [14, 16]. First, dynamic objects are extracted from the original video and then realigned in time *via* minimization of a carefully selected cost function; no spatial transformation is allowed thus preventing the total loss of context that often takes place when both spatial and temporal manipulations are allowed. The experimental results shown on the authors' web page are very convincing and demonstrate viability of the approach. The overall method, however, is complex, requiring several computational stages involving object detection, object queuing, background fusion, tube selection, and tube stitching.

The video condensation approach we propose here is closest to video synopsis as we allow spatial segments from different frames to be combined in the output video while permitting only temporal transformations (spatial location of dynamic events is preserved). At the same time, our approach does not require object extraction; instead, we use simple background subtraction to guide the video condensation process. We introduce the concept of a ribbon in space-time video volume as a novel extension of the concept of a seam in 2-D images used for content-aware image resizing. We also introduce a ribbon flex-parameter to trade off the video condensation ratio and anachronism of events, and develop a hierarchical sliding-window approach to process endless video in multiple passes.

3 Seam carving for image resizing

A novel technique called seam carving was recently introduced in [2] for content-aware image resizing. At a high level, it is a method to change the size of an image by moderate amounts by recursively reducing or increasing the horizontal/vertical dimension one row/column at a time. This one row/column reduction or increase is performed by respectively deleting or inserting a set of pixels which is referred to as a seam. Seams are associated with costs which can reflect scene content, with larger costs associated with more important content. The idea is to delete or insert seams with the least cost in a recursive manner, re-computing costs as needed, till the desired aspect ratio is achieved. The recursive procedure together with the cost function tries to minimize the amount of content degradation caused by the insertion or deletion of seams.

At the time of writing this paper, we learned about the recent work [17] where this procedure was extended to change the *spatial* size of video frames and was referred to as improved seam carving for video retargeting. In contrast to seam carving for image resizing or video retargeting, in video condensation we want to keep the spatial size of the video frames intact and instead only *reduce* the temporal dimension of the video while preserving important *events* and their *relative timings*. To do this, we extend the seam carving idea to video in a manner which is orthogonal to the extension of seam carving to video retargeting. The concluding section of reference [17] briefly comments on ongoing and planned efforts to apply seam carving ideas to video summarization. It also cites the recent reference [4] where this has been tried. Ours is an independent parallel work which differs from these recent developments in a number of ways. We will point out the essential differences and similarities of our method with these recent developments in the subsequent sections.

Since we build upon the key idea of seam carving for image resizing, we now describe the notion of a seam in more detail. Our goal is to apply this to video condensation so our discussion is focused on seam deletion instead of seam insertion.

(a)

(b)

(c)

(d)

Figure 1: Illustration of the principle and benefits of seam carving: (a) original image with horizontal and vertical seams superposed (seams are thickened for visibility); (b) cropped image; (c) uniformly-scaled image; and (d) seam-carved image [2].

In a rectangular image which is W pixels wide and H pixels tall, a vertical seam is a set of pixels which

1. extend from the top to the bottom of the image,
2. have different vertical coordinates, and
3. are path-connected.

Thus, a vertical seam is, crudely speaking, a vertically oriented curve which partitions the image into left and right parts. Due to properties 1) and 2), deleting the pixels in

a vertical seam reduces the width of an image by exactly one column. The additional connectivity property 3) is required for a seam to ensure that deleting a seam does not introduce unnatural discontinuities which can lead to high content distortion. A vertical seam can be formally defined as a set of pixels $(x(y), y)$, $y = 1, \ldots, H$, where $x(y)$ is a function with range $1, \ldots, W$, and the property that $|x(y+1) - x(y)| \leq 1$ for all $y = 1, \ldots, (H-1)$. Thus, $x(y)$ describes the graph of the vertically oriented curve which defines a seam. A vertical seam is shown in Fig. 1(a) in red. A horizontal seam can be similarly defined as a set of pixels $(x, y(x))$, $x = 1, \ldots, W$, where $y(x)$ is a function with range $1, \ldots, H$, and the property that $|y(x+1) - y(x)| \leq 1$ for all $x = 1, \ldots, (W-1)$. A horizontal seam is shown in Fig. 1(a) in yellow. Deleting the pixels in a horizontal seam reduces the height of an image by exactly one row.

The key idea of seam deletion for content-aware image down-sizing is to associate content-aware costs with seams and then recursively remove seams one at a time, recalculating the costs if needed, until the image is reduced to the desired size. Typical content-aware costs used in [2] are based on weighted intensity gradients. The total number of vertical (respectively horizontal) seams grows exponentially with the height (respectively width) of the image. Thus, an exhaustive search for a minimum-cost seam for even moderate image sizes is computationally prohibitive. To overcome this difficulty, [2] uses cost functions which are additive over the pixels in a seam, e.g., the sum of the absolute values of the local intensity gradients over all pixels in a seam. An additive cost function together with the structure of a seam make it possible to reformulate the search for a minimum-cost seam in terms of a dynamic programming algorithm which is guaranteed to find a minimum-cost seam and has a computational complexity which is linear in the number of pixels in the image. To avoid repetition, we refer the reader to [2] for details of the dynamic programming algorithm for seam carving. In Sec. 4.3 we explain in detail the analogous efficient search method for video condensation by ribbon carving.

Fig. 1 illustrates unique properties of image down-sizing my means of seam carving *vis-à-vis* cropping and uniform scaling. Note that while cropping removes pertinent

parts of the original image, uniform scaling distorts object appearance (incorrect aspect ratio). On the other hand, seam carving largely preserves the pertinent content while avoiding excessive geometric distortions.

4 Ribbon carving for video condensation

Motivated by seam carving for image resizing, we now introduce the key ideas of ribbon carving for video condensation.

4.1 Video seams

Suppose that we have a segment of N consecutive video frames where each frame is W pixels wide and H pixels tall. After condensation, we want to end up with a new segment of N' consecutive frames of the same width and height but with $N' \leq N$. The condensation procedure should hopefully preserve all the important events, e.g., moving vehicles and walking pedestrians, and their relative timings while not degrading the quality of the perceived video in a noticeable way. We want to do this by mimicking the idea of seam deletion for image down-sizing. In order to do so, we need to define the 3D counterpart of a 2D seam within the spatio-temporal volume of a video segment. We call this counterpart a (3D) video seam.

Deleting the pixels in a video seam should:

(a) preserve the spatial size of the resulting video segment,

(b) reduce the number of frames in the video segment by exactly one, and

(c) prevent significant event distortion in either space or time.

Thus, a video seam is, crudely speaking, a connected surface within the spatio-temporal volume of a segment of video frames which partitions the video into "past" and "future" connected regions. The spatial size of all the video frames remains unchanged. In contrast, in the video retargeting application studied in [17], a video seam

partitions the spatio-temporal volume into left and right or top and bottom regions and changes the *spatial* size of the video frames. Requirements (a)–(c) above can, for example, be formalized by defining a video seam as a set of pixels $(x, y, t(x, y))$, $x = 1, \ldots, W$, $y = 1, \ldots, H$, where $t(x, y)$ is a function with range $1, \ldots, N$, and the property that $|t(x, y) - t(x', y')| \leq 1$ for all pairs of spatial coordinates (x, y) and (x', y') for which $|x - x'| \leq 1$ and $|y - y'| \leq 1$. Thus, $t(x, y)$ describes the graph of a surface, within the spatio-temporal volume of a video segment, which defines a video seam. Since $t(x, y)$ is a function, no two pixels in a video seam have the same spatial coordinates. This property ensures that deleting the pixels in a video seam reduces the temporal dimension of the video segment by exactly one. The set of pixels forming a video seam are also path-connected in the spatio-temporal volume of the video segment. This property ensures that seam deletion does not lead to significant distortion of events in space and time.

It should be noted that seam deletion is, in general, a highly nonuniform and nonlinear downsampling operation. Every pixel in a condensed video which is produced by successive seam deletions comes from some pixel in the original video. The intensity and color of the pixels in the original video are not modified. This should be contrasted with standard image and video resizing where the intensity of the pixels are modified by anti-alias filtering prior to downsampling. Thus, the condensed video will contain a strict subset of pixels from the original video but rearranged in some way. In the common usage of the term video summarization, the intensity and color values of pixels in the original video are allowed to be modified; whereas, in what we refer to as video condensation, this is not allowed. This is a subtle but important technical difference between video summarization and video condensation.

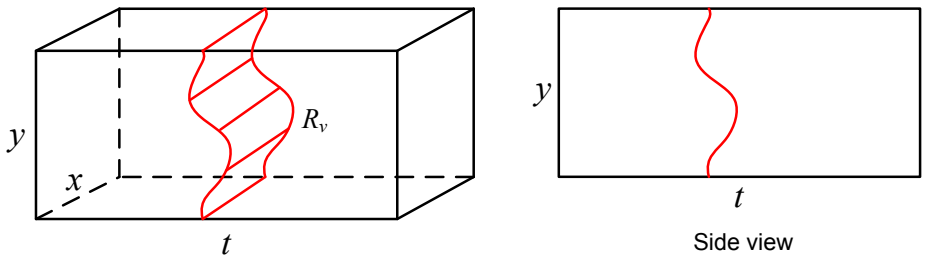
The key idea of seam carving for video condensation is to associate reliable activity-aware costs with seams and then recursively remove seams one at a time, recalculating the costs if needed, until a certain user-defined stopping criterion is met. A very crude indicator of activity is the spatio-temporal *intensity* or *color* gradient. This is similar in spirit to the cost used in seam carving for image resizing in [2].

A cost function based on intensity gradient is used in [4] and is mentioned in [17] under ongoing research on video summarization. However, as discussed in Sec. 5, cost functions which are based on intensity or color gradients typically produce significant spatio-temporal distortion in the condensed video. In contrast, we advocate the use of more sophisticated measures of activity based on state-of-the-art background subtraction algorithms. These costs preserve the shapes of moving objects and introduce minimal spatio-temporal distortion in the condensed video.

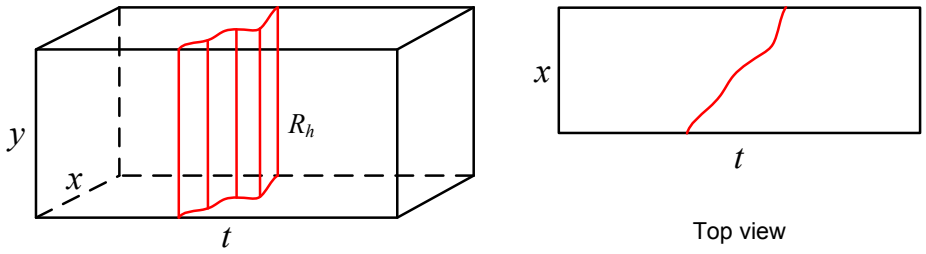
In general, an exhaustive search for a minimum-cost video seam, for even moderate video sizes, is computationally prohibitive. If the cost function is additive over the pixels in a video seam then it turns out that this problem can be converted to a max-flow min-cut problem on an appropriately-defined directed graph with fictitious source and sink vertices. This is referred to as the graph cut algorithm in the computer vision literature [3]. Reference [17] explains how this can be done in the context of video retargeting. Essentially, the pixels in the video form the vertices of a graph and the edge capacities and directions are determined by the cost function and the topological constraints of a video seam. The graph cut algorithm finds a minimum-cost video seam with a computational cost which is typically *quadratic* in the number of pixels in the video segment [17], i.e, proportional to $(NWH)^2$. Although this is polynomial in the number of pixels, it is still prohibitive for typical video sizes. To make the search more tractable, one approach is to take recourse to an approximate method as in [17]. Another approach is to impose additional structure on a video seam to reduce the range of search possibilities. We take the latter approach by introducing the notion of horizontal and vertical ribbons in the next subsection. This allows us to reuse the dynamic programming algorithm for image seam carving to find a minimum-cost ribbon with a computational complexity which is proportional to $N(W + H)$ (see Sec. 4.3 for justification).

4.2 Horizontal and vertical ribbons

A vertical video slice at position x_0 is a set of pixels (x_0, y, t) which have the same horizontal coordinate x_0. Similarly, a horizontal video slice at position y_0 is a set of pixels which have the same vertical coordinate y_0. The intersection of a vertical video slice at location x_0 with a 3D video seam defines a curve $(x_0, y, t(x_0, y))$ which forms a vertically oriented 2D seam within the vertical video slice at position x_0. As x_0 ranges from 1 through W, the vertical 2D seams formed by the intersection of consecutive vertical video slices with the 3D video seam will be different in general. However, if the vertical 2D seams so formed are the same for all values of x_0, that is, $t(x, y)$ does not depend on x, then the 3D video seam is essentially a vertically oriented 2D seam within a vertical video slice extended horizontally within the video segment. We call such a 3D video seam a *vertical ribbon*. A vertical ribbon is shown in Fig. 2(a).



(a)



(b)

Figure 2: Schematic illustration of: (a) vertical ribbon, and (b) horizontal ribbon.

Formally, a vertical ribbon is a set of pixels $(x, y, t(y))$, $x = 1, \ldots, W$, $y =$

$1, \ldots, H$, where $t(y)$ is a function of only y with range $1, \ldots, N$, and the property that $|t(y+1) - t(y)| \leq 1$ for all $y = 1, \ldots, (H-1)$. In an analogous manner we define a horizontal ribbon as a horizontally-oriented seam within a horizontal video slice extended vertically within the video segment. Formally, a horizontal ribbon is a set of pixels $(x, y, t(x))$, $x = 1, \ldots, W$, $y = 1, \ldots, H$, where $t(x)$ is a function of only x with range $1, \ldots, N$, and the property that $|t(x+1) - t(x)| \leq 1$ for all $x = 1, \ldots, (W-1)$. A horizontal ribbon is shown in Fig. 2(b).

4.3 Finding minimum-cost ribbons

In ribbon carving, we associate activity-aware costs with ribbons and then find the least-cost ribbon among the set of all horizontal and vertical ribbons to delete. Let C be a cost function which associates with each pixel having coordinates (x, y, t) a cost denoted by $C(x, y, t)$. If R denotes a horizontal or vertical ribbon then the cost of the ribbon is given by

$$C(R) \;=\; \sum_{(x,y,t) \in R} C(x, y, t).$$

In other words, the cost of a ribbon is equal to the sum of the costs of the individual pixels which make up the ribbon. This additive structure of the ribbon cost function together with the spatio-temporal structure of a ribbon make it possible to find minimum-cost horizontal and vertical ribbons in a computationally efficient way. We illustrate the procedure for vertical ribbons. An identical procedure can be used for finding a minimum-cost horizontal ribbon. Once minimum-cost horizontal and vertical ribbons have been found, whichever ribbon has smaller cost will be selected for deletion.

First, note that a vertical ribbon is completely determined by a vertical seam in a vertical video slice. In other words, there is a one-to-one correspondence between vertical seams on images which are N pixels wide and H pixels tall and vertical video ribbons (see Fig. 2(a)). Let $R_v = \{(x, y, t(y)) : x = 1, \ldots, W,\ y = 1, \ldots, H\}$ be a

vertical ribbon. The cost of R_v is given by

$$\begin{aligned} C(R_v) &= \sum_{(x,y,t)\in R_v} C(x,y,t) \\ &= \sum_{y=1}^{H} \left[\sum_{x=1}^{W} C(x,y,t(y)) \right] \\ &= \sum_{y=1}^{H} \tilde{C}_v(t(y),y) \end{aligned}$$

where $\tilde{C}_v(t,y) := \sum_{x=1}^{W} C(x,y,t)$ is a 2D cost function associated with a vertical video slice. Thus, finding a minimum-cost vertical ribbon is equivalent to finding a minimum-cost vertical seam $(t(y),y), y = 1,\ldots,H$, in a vertical video slice "image" which is N pixels wide and H pixels tall, and where the per-pixel cost is given by $\tilde{C}_v(t,y)$. A minimum-cost vertical seam can now be found by the dynamic programming algorithm used for 2D image seam carving. The algorithm is fairly intuitive and works as follows. Working from the bottom to the top of the image, for $i = 1,\ldots,N$, and $y = 1,\ldots,H$, let $t_i^*(y)$ denote the least-cost vertical path which begins at the bottom of the image ($y = 1$) and ends at pixel with coordinates (i,y). Let $C_i^*(y)$ be the cost of $t_i^*(y)$. Since seams are path-connected, ignoring image boundaries, at each horizontal position i there are three possible choices for extending the best paths up to height y to position $(i,y+1)$ at height $(y+1)$. The three choices are the paths $t_{i-1}^*(y)$, $t_i^*(y)$, and $t_{i+1}^*(y)$ with costs $C_{i-1}^*(y)$, $C_i^*(y)$, and $C_{i+1}^*(y)$ respectively. Clearly, the least-cost path among these three should be chosen to be extended to position $(i,y+1)$. If two or more paths have the same cost, then we can choose any of them. The cost of the extended-path is updated by adding the cost $\tilde{C}_v(i,y+1)$ at position $(i,y+1)$ to the cost of the path before extension. Initially, $C_i^*(1) = \tilde{C}_v(i,1)$. Upon reaching the top of the image ($y = H$) the vertical path with the least cost among the N vertical paths at height H will be a minimum-cost vertical seam. There is one addition operation and two cost comparisons performed for each position (i,y). Hence the computational cost of this dynamic programming procedure is proportional to NW. Similarly, the computational cost for finding a minimum-cost vertical ribbon is

proportional to NH. Hence the total computational cost is of the order of $N(W+H)$.

4.4 Cost functions

In video condensation, we would like to decrease the temporal dimension while preserving events and their relative timings. The most significant events are those due to the motion of objects. Object motion expresses itself in the form of intensity and color changes at the resolution of an individual pixel. Thus, a good pixel-level cost function should reliably indicate motion-related activity.

A crude indicator of such activity is the local intensity gradient. If I'_x, I'_y and I'_t denote local estimates of horizontal, vertical and temporal derivatives of intensity I at (x, y, t), respectively, one possible cost function is the length of intensity gradient in space-time (3-D):

$$C(x,y,t) = \sqrt{(I'_x(x,y,t))^2 + (I'_y(x,y,t))^2 + (I'_t(x,y,t))^2}.$$

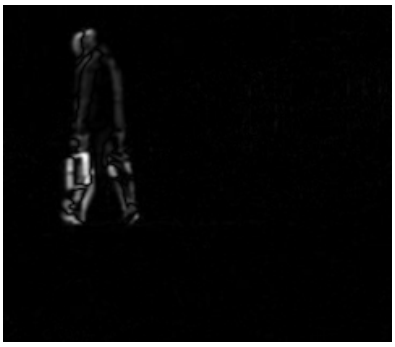
Each derivative can be estimated, for example, by a 3-D Sobel operator that computes a difference along one axis and performs averaging along the remaining axes. This type of cost was proposed for video carving recently [4]. Note, however, that due to the contribution of spatial derivatives this cost will be high not only in moving areas but also in textured still areas (Fig. 3.b), thus potentially preventing carving through those areas. We will discuss this further in Sec. 5.

The impact of static features on the overall cost can be mitigated by considering only the temporal derivative. Thus, we consider a cost function based on its magnitude:

$$C(x,y,t) = |I'_t(x,y,t)|.$$

As shown in Fig. 3.c, a high cost is present only in the area of the moving pedestrian. Note, however, that this cost does not capture accurately the interior of the moving pedestrian; only the boundaries are captured accurately. Thus, the low cost in the interior may encourage a ribbon cut through a moving object potentially leading to distortions.

(a)



(b) (c) (d)

Figure 3: (a) Original video frame from the *Sidewalk* sequence, and (b) continuous cost based on magnitude of 3-D luminance gradient $(x-y-t)$ as proposed in [4], (c) continuous cost based on magnitude of temporal luminance derivative (Sec. 4.4), (d) binary cost based on activity (motion) labels resulting from background subtraction [10].

The above cost functions based on estimates of local intensity gradients are similar in spirit (but not the same) to those used in [2] and [17]. However, they are not very reliable indicators of object motion because, at any given position, gradients can change over time without any object motion due to a variety of reasons. This leads to significant spatio-temporal distortion in the condensed video (see Sec. 5). More robust indicators of motion-related activity at the pixel level have been developed recently using background subtraction algorithms [5, 10, 18, 19]. While the

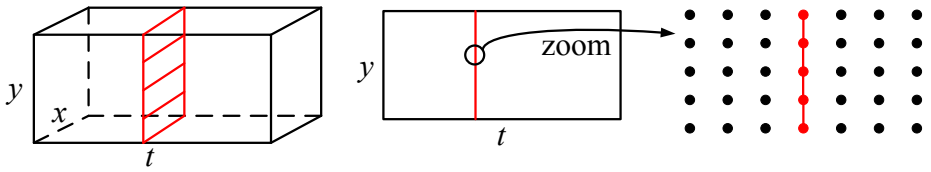
details vary, a typical background subtraction algorithm learns a statistical model (a prior probability distribution) for the background intensity and color at a given position using samples at that location from previous frames. A binary decision, with 0 denoting background and 1 denoting foreground (motion-related activity), is then made using a likelihood ratio test based on the learned prior and the observed values. Often, this is combined with spatial priors to incorporate object connectivity constraints while making background/foreground decisions. In our experiments, we used the background subtraction algorithm developed in [10]. This leads to a binary cost shown in Fig. 3.d.

There is one important conceptual difference between the intensity-based cost functions used in [2,4,17] and an activity-aware cost function that we are advocating here. In [2,4,17], when a seam is removed, the intensity gradients in the vicinity of the seam are typically altered and some local distortion is introduced. Recomputing the intensity gradients after each seam removal is in keeping with the philosophy that successive seam removals should introduce the least amount of distortion. In contrast, in video condensation the focus is on preserving all those pixels in the original video which correspond to motion-related activity. This is a property of the original video. Each ribbon removal may introduce some spurious "activity" in the vicinity of a ribbon but these are artifacts of the condensation process. As such, they are not intrinsic to the original video and should not be erroneously declared active and preserved. Hence, motion-related activity should not be recomputed on the condensed video as successive ribbons are removed but only determined once from the original video. Thus, in addition to providing a more reliable indicator of motion-related activity than a gradient-based cost function, a side-advantage of our cost function is that it does not have to be recomputed after every ribbon deletion because the true motion-based activity is a property of the original video that needs to be preserved in the condensed video.

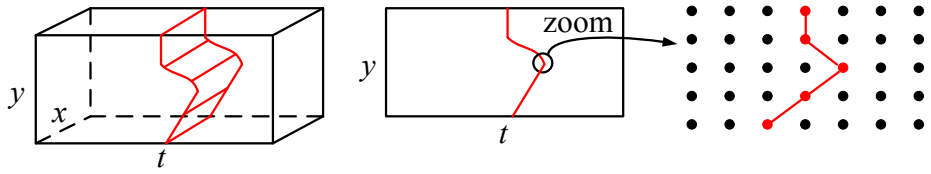
4.5 Stopping criteria and ribbon flexibility

In order to condense a given video segment by ribbon carving, minimum-cost ribbons need to be removed one at a time till a user-defined stopping criterion is met. One possible choice for the stopping criterion is the target condensation ratio. This is the approach used in [2] and [17] for image and video resizing respectively and in [4] for video summarization. A drawback of such a criterion is that it is not adaptive to the irregular, bursty nature of scene activity and there is no obvious way to use such a criterion for condensing an "endless" stream of video. In the context of processing endless video, the choices for a stopping criterion are particularly unclear for intensity-based cost functions. For activity-based cost functions, however, there are several natural choices for a stopping criterion which is activity-adaptive and also allows for an automatic control of the condensation ratio. For instance, for the binary cost function based on background subtraction, which we use, a possible stopping criterion is to check if the cost of a minimum-cost ribbon is strictly positive. In other words, ribbons are deleted until it is no longer possible to find a minimum-cost ribbon which has exactly zero cost. This will ensure that every pixel which has a positive cost, i.e., all pixels in moving objects, will be preserved in the condensed video. If in some application it is not important to preserve every moving-object pixel but only a significant fraction, other stopping criteria can be used. For instance, one could remove ribbons until the cumulative cost of all ribbons removed exceeds some user-defined fraction of the total number of foreground pixels in the segment of video.

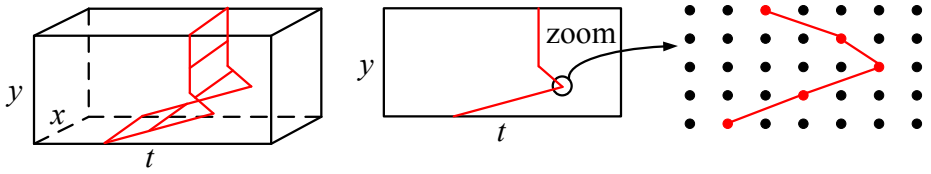
For any stopping criterion, the extent to which a video segment can be condensed is limited by the structural properties of a ribbon. The one-pixel neighborhood connectivity requirement of a ribbon restricts the extent to which a ribbon can flex around moving-object "tunnels" in the 3D space-time volume without cutting them. This restricts the condensation ratio. To achieve higher, tunable, condensation ratios, we introduce the notion of an integer-valued ribbon "flex" parameter ϕ which controls the maximum deviation or jump from connectivity. Formally, we define a vertical ribbon with a flex parameter ϕ as a set of pixels $(x, y, t(y))$, $x = 1, \ldots, W$,



(a) $\phi = 0$



(b) $\phi = 1$



(c) $\phi = 2$

Figure 4: Schematic illustration of vertical ribbons with different degrees of flexibility parameterized by the flex parameter ϕ: (a) $\phi = 0$; (b) $\phi = 1$; and (c) $\phi = 2$.

$y = 1, \ldots, H$, where $t(y)$ is a function of only y with range $1, \ldots, N$, and the property that $|t(y+1) - t(y)| \leq \phi$ for all $y = 1, \ldots, (H-1)$. Similarly, a horizontal ribbon with flex parameter ϕ is a set of pixels $(x, y, t(x))$, $x = 1, \ldots, W$, $y = 1, \ldots, H$, where $t(x)$ is a function of only x with range $1, \ldots, N$, and the property that $|t(x+1) - t(x)| \leq \phi$ for all $x = 1, \ldots, (W-1)$. The parameter ϕ indicates the degree of flexibility of a ribbon with larger values corresponding to greater flexibility and less connectivity. If $\phi = 0$ then the ribbon coincides with a frame. Ribbons originally defined in Sec. 4.2 are included as a special case corresponding to $\phi = 1$. Fig. 4 schematically illustrates vertical ribbons of different degrees of flexibility.

While larger values of ϕ allow higher condensation ratios to be realized, this comes

at the price of greater event anachronism, i.e., the relative ordering of events in space and time can become more distorted. However, we can *carefully control the trade-off* between condensation ratio and event anachronism by tuning the flex parameter ϕ. For example, when $\phi = 0$, ribbon removal is the same as frame removal. Removing a frame does not change the relative timings of events. However, the ability to condense video in this manner is limited. For the same stopping criterion, as we increase the value of ϕ, the condensation ratio increases. However, after a point the improvements in the condensation ratio become marginal. In our experiments (see Table 5 in Sec. 6), we found that values of ϕ larger than 4 produce marginal improvements in the condensation ratio when the stopping criterion insists on preserving all active pixels (pixels with positive cost). Fig. 5 shows minimum-cost horizontal ribbons of different degrees of flexibility found by our dynamic programming algorithm for a binary cost function based on the background subtraction algorithm in [10]. In the figure, pixels with cost one (active foreground) are shown in blue and correspond to moving objects. Moving objects trace object tunnels in the 3D space-time while ribbons cut through space-time in-between these tunnels.

4.6 Processing "endless" video and the overall algorithm

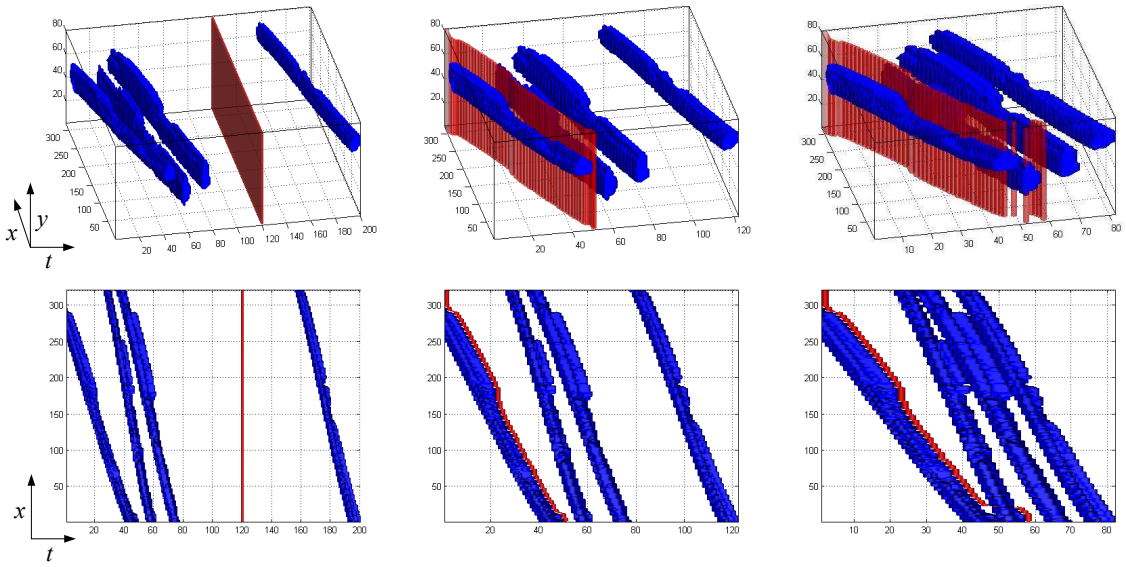
Up to now we described how to condense a given segment of N video frames. When we have hours of video to condense, it is impractical to load all the video frames at the same time. To overcome this difficulty, we adopt a sliding-window condensation procedure described below.

The key observation is that a horizontal ribbon with flex parameter ϕ cannot span more than $(\phi W - \phi + 1)$ frames and a vertical ribbon with flex parameter ϕ cannot span more than $(\phi H - \phi + 1)$ frames. Thus, a horizontal or vertical ribbon with flex parameter ϕ cannot span more than $M_\phi := (\phi \max(W, H) - \phi + 1)$ frames. For a fixed value of ϕ and a stopping criterion, we begin with a segment of N frames and keep removing the minimum-cost horizontal/vertical ribbons till the stopping criterion is met, i.e., there are no more horizontal or vertical ribbons of flex parameter ϕ which

can be removed without violating the stopping criterion for the segment. Let N' be the number of frames left at the time of stopping and let them be re-indexed as $1, \ldots, N'$, with smaller frame numbers coming "earlier" than larger frame numbers in the condensed time-scale. If $N'' := (N' - M_\phi)$ is strictly positive then we can save the "last" N'' frames in the block of N' frames as condensed video, push the $(N' - N'') = M_\phi$ frames to the "back" of the block and and read in $(N - N' + N'') = (N - M_\phi)$ new frames to the "front" to create a new segment of N frames to condense. The reason we can save the N'' frames is because we know that no ribbon which begins in this set of frames can reach beyond frame number N' and at the time of stopping there are no additional ribbons which can be removed from these N'' frames. If $N'' \leq 0$ then we cannot save any frames and we only read in $(N - N')$ new frames to the "front" to create a new segment of N video frames to be condensed. Although N can be any number larger than M_ϕ, for implementation efficiency we use $N = 2M_\phi$.

We now describe our overall video condensation algorithm based on deleting ribbons of increasing degrees of flexibility using the above sliding window procedure. We are given the following: 1) a video sequence with spatial dimensions $W \times H$, 2) an activity-aware cost function $C(x, y, t)$ which can be either precomputed from the video or can be updated during the condensation process, 3) a stopping criterion, and 4) a maximum value of ϕ say $\phi_{\max}$. Video condensation is performed in multiple levels. We start the condensation with $\phi = 0$ (level-zero) and use the sliding window procedure described above with $N = 2M_\phi$. This involves condensing a segment of N frames by removing minimum-cost ribbons of flex parameter ϕ one at a time till the stopping criterion is met, and updating the block of frames by saving condensed frames and reading in new frames. This generates a level-zero condensed video sequence where only frames have been removed ($\phi = 0$), so there is no event anachronism. In the next level, we increase ϕ by one and repeat the sliding window condensation procedure on the condensed video output of the previous level. This produces a level-one condensed video. This is repeated through multiple levels of condensation till $\phi = \phi_{\max}$. The condensation in multiple levels can be carried out in

parallel with suitable frame buffering and delays. We start with $\phi = 0$ and gradually increase it to $\phi_{\max}$ so that event anachronism can be kept to a minimum. After level-zero we have essentially removed as many frames as we possibly can without violating the stopping criterion. To condense further we have no choice but to introduce event anachronism. So we choose the next smallest value of ϕ available and proceed. The results in Table 5 of Sec. 6 show how much more condensation is possible after each level of processing. This quantifies the "value" of higher degrees of flexibility in terms of their contribution to the condensation.



$\phi = 0$ $\phi = 1$ $\phi = 6$

Figure 5: Object tunnels (blue) and minimum-cost horizontal ribbons (red) of different degrees of flexibility obtained from the *Highway* sequence.

5 Experimental results

We have tested our video condensation algorithm on a number of sequences captured by network cameras on Boston University campus. Here, we report results for three sequences: *Highway*, *Overpass* and *Sidewalk*. First, we applied background subtrac-

(a) (b) (c)

Figure 6: Sample frame from condensed video produced using: (a) continuous cost based on magnitude of 3-D luminance gradient [4], (b) continuous cost based on magnitude of temporal luminance derivative, and (c) binary cost based on activity labels.

tion to each sequence in order to compute an activity map for each frame. We used an extension of the classical background subtraction algorithm of ElGammal *et al.* [5]. In addition to a non-parametric background probability distribution model (computed over 50 preceding frames), we used a non-parametric, local-in-space foreground model to improve the method's discrimination sensitivity and a Markov prior model to improve the spatial coherence of activity labels [10]. A typical activity map obtained for the *Sidewalk* sequence is shown in Fig. 3.d. Although far from perfect, it nevertheless accurately portrays dynamics occurring in this frame.

First, we compare the efficacy of the three costs discussed in Sec. 4.4, namely those based on spatio-temporal gradients, temporal gradients, and activity labels. Since in all our experiments we opted for zero-activity-loss stopping criterion (no single moving pixel removed), it is unclear what stopping criterion to use for the two gradient costs in order that the comparison be fair. Therefore, we first ran the activity-cost algorithm and found how many ribbons to remove for each flexibility level. Then, we removed exactly this many lowest-cost ribbons in each gradient-based algorithm. Clearly, this is the most favorable scenario for the gradient-based condensation algorithms. Fig. 6 compares the results obtained using the three different costs. The frame condensed

Table 1: Experimental setup and video condensation ratios attained for the three sequences tested with no distortion (no moving pixel carved out).

Sequence	*Highway*		*Overpass*		*Sidewalk*	
Length	23,950 frames		23,950 frames		7,950 frames	
Resolution	320×80 pixels		160×60 pixels		240×208 pixels	
Sliding window	640 frames		320 frames		480 frames	
Length of condensed video sequence and cumulative condensation ratio						
$\phi = 0$	8,114	2.95:1	11,880	2.02:1	4,462	1.78:1
$\phi = 1$	3,186	7.52:1	9,174	2.61:1	3,519	2.26:1
$\phi = 2$	3,031	7.90:1	8,609	2.78:1	3,445	2.31:1
$\phi = 3$	2,970	8.06:1	8,331	2.87:1	3,431	2.32:1

using the spatio-temporal gradient cost (Fig. 6.a) exhibits significant distortions; the body of the pedestrian is largely missing. This is due to the fact that the cost (1) is significant if either of the three intensity derivatives (horizontal, vertical, or temporal) is large. Thus, static background features exhibit high cost (Fig. 3.b) preventing cuts through them. Incidentally, the pedestrian's jacket has uniform intensity (low cost) thus inviting cuts. This results in most of pedestrian's body disappearing, and also in a severe anachronism as parts of the pedestrian body appear twice in this frame (right-hand side). The frame condensed using the temporal-gradient cost (Fig. 6.b) fares better since static features have low cost (Fig. 3.c) and can be easily removed. This eliminates the severe anachronism observed in Fig. 6.a, however, it does not prevent the strobing effect (multiple appearances of a moving object in very close succession). This effect is due to the fact that the temporal gradient is incapable of assigning a high cost to the interior of a uniform-intensity object (Fig. 3.c) thus permitting a ribbon cut. The result based on activity cost (Fig. 6.c) is void of any major distortions. This is due to the zero-activity-loss stopping criterion and label contiguity (Fig. 3.d); had labels been more fragmented, a cut through the moving

object would have been possible. Clearly, a spatially-coherent, movement-accurate label field is essential for successful ribbon carving.

In the remaining experiments, we have used only the activity cost and tolerated no loss of activity in the condensed video. Table 5 gives details of the experimental setup and shows the cumulative condensation ratios attained at each level of ribbon flexibility. First, note that condensation ratios attained are moderate due to the fact that no activity loss is permitted, unlike in prior methods [14]. Secondly, note that for the three sequences tested, the relative gains diminish as the ribbon flexibility increases. This was to be expected since the object tunnels are progressively pushed closer together leaving less and less free space in-between (Fig. 5). Compared to the *Overpass* and *Sidewalk* sequences, the condensation ratios for the *Highway* sequence are higher and the gains due to higher degrees of ribbon flexibility are more significant. This is so because there are longer periods of inactivity *and* there is less variability in the motion because it is unidirectional and all the vehicles have very similar velocities. This creates fairly parallel, sparsely-spaced object tunnels. In the *Overpass* and *Sidewalk* sequences, few frames are void of moving objects, and these objects have different, even opposite, velocities. This leaves little space for ribbons. Finally, in all three sequences we have observed that horizontal ribbons are removed far more frequently than vertical ribbons. This is due to the fact that in all three sequences the object motion is predominantly horizontal thus creating structural gaps horizontally between object tunnels (Fig. 5) that are exploited by horizontal ribbons. The contrary would be true had object motion been vertical.

Figs. 7-9 show the result of ribbon carving for the three tested sequences, respectively. In each figure, three frames from the original video sequence are shown, each with different objects (cars, people), together with one frame from the condensed video that includes all these objects. Note that the object order is maintained while no strobing or ghosting effects are introduced. In order to fully appreciate the performance of our condensation algorithm, we invite the reader to view the condensed videos on our web page [1].

6 Concluding remarks

We have presented a novel approach to efficient browsing of surveillance video. Our method for condensing video is straightforward to implement, computationally efficient, and produces high-fidelity condensed video, both in terms of moving object integrity and relative timing. This high fidelity, however, can be traded for a better condensation ratio thanks to a flexible-ribbon model that we proposed. Our best-performing variant of video condensation is based on activity (motion) cost rather than intensity-gradient cost, and thus a reliable detection of motion labels, e.g., by means of background subtraction, is paramount to the effectiveness of our approach. Our ribbon carving is very effective for typical surveillance video with multiple, similarly-moving objects (direction and speed), but cannot condense scenes with multiple objects moving at very different speeds and/or directions. In this case, the removal of video seams that flex both horizontally and vertically (Sec. 4.1), by minimizing an activity-aware cost, could potentially improve the condensation ratio further. An alternative to the use of more flexible seam surfaces for improving the condensation ratio is to allow for a graceful loss of activity. These are interesting directions for future research.

Acknowledgment

The authors want to thank Dr. P.-M. Jodoin, Département d'Informatique, Université de Sherbrooke, Canada, for reference [16] and helpful comments.

References

[1] http://iss.bu.edu/jkonrad/Research/VidCon/vidcon.html.

[2] S. Avidan and A. Shamir, "Seam carving for content-aware image resizing," *ACM Trans. Graph.*, vol. 26, no. 3, 2007.

[3] Y. Boykow and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 26, pp. 1124–1137, Sept. 2004.

[4] B. Chen and P. Sen, "Video carving," in *EUROGRAPHICS'08*, 2008.

[5] A. Elgammal, R. Duraiswami, D. Harwood, and L. Davis, "Background and foreground modeling using nonparametric kernel density for visual surveillance," *Proc. IEEE*, vol. 90, pp. 1151–1163, 2002.

[6] A. Hampapur, "Smart video surveillance for proactive security," *IEEE Signal Process. Mag.*, vol. 25, no. 4, pp. 136–134, 2008.

[7] M. Irani, P. Anandan, J. Bergen, R. Kumar, and S. Hsu, "Efficient representations of video sequences and their applications," *Signal Process., Image Commun.*, vol. 8, pp. 327–351, May 1996.

[8] H.-W. Kang, Y. Matsuhita, X. Tang, and X.-Q. Chen, "Space-time video montage," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, pp. 1331–1338, June 2006.

[9] J. Konrad, "Videopsy: Dissecting visual data in space-time," *IEEE Comm. Mag.*, vol. 45, pp. 34–42, Jan. 2007.

[10] J. McHugh, J. Konrad, V. Saligrama, and P.-M. Jodoin, "Foreground-adaptive background subtraction," *IEEE Signal Process. Lett.*, Sept. 2008 (submitted).

[11] J. Nam and A. Tewfik, "Video abstract of video," in *IEEE Workshop on Multimedia Signal Proc.*, pp. 117–122, 1999.

[12] J. Oh, Q. Wen, J. Lee, and S. Hwang, "Video abstraction," in *Video Data Mangement and Information Retrieval* (S. Deb, ed.), ch. 3, pp. 321–346, Idea Group Inc. and IRM Press, 2004.

[13] N. Petrovic, N. Jojic, and T. Huang, "Adaptive video fast forward," *Multimedia Tools Appl.*, vol. 26, pp. 327–344, Aug. 2005.

[14] Y. Pritch, A. Rav-Acha, and S. Peleg, "Non-chronological video synopsis and indexing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 30, Nov. 2008 (to appear).

[15] A. Rav-Acha, Y. Pritch, D. Lischinski, and S. Peleg, "Dynamosaicing: Mosaicing of dynamic scenes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 29, no. 10, pp. 1789–1801, 2007.

[16] A. Rav-Acha, Y. Pritch, and S. Peleg, "Making a long video short: Dynamic video synopsis," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, pp. 435–441, June 2006.

[17] M. Rubinstein, A. Shamir, and S. Avidan, "Improved seam carving for vieo retargetting," *ACM Trans. Graph.*, vol. 27, no. 3, 2008.

[18] Y. Sheikh and M. Shah, "Bayesian modeling of dynamic scenes for object detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 27, no. 11, pp. 1778–1792, 2005.

[19] C. Stauffer and E. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 8, pp. 747–757, 2000.

[20] O. Tuzel, F. Porikli, and P. Meer, "Pedestrian detection via classification on Riemannian manifolds," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 30, pp. 1713–1727, Oct. 2008.

[21] J. Vlahos, "Welcome to the Planopticon," *Popular Mechanics*, pp. 64–69, Jan. 2008.

[22] M. Yeung and B.-L. Yeo, "Video visualization for compact presentation and fast browsing of pictorial content," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 771–785, Oct. 1997.

(a) Original frame #12,852.

(b) Original frame #12,933.

(c) Original frame #12,947.

(d) Condensed frame #1,808.

Figure 7: Result of ribbon carving with $\phi_{max} = 2$ for the *Highway* sequence: (a-c) three original frames; (b) frame from the condensed sequence comprising objects from different original frames (times).

(a) Original frame #9,757.

(b) Original frame #10,065.

(c) Original frame #10,557.

(d) Condensed frame #4,445.

Figure 8: Result of ribbon carving with $\phi_{max} = 2$ for the *Overpass* sequence: (a-c) three original frames; (b) frame from the condensed sequence comprising objects from different original frames (times).

(a) Original frame #582.

(b) Original frame #1,495.

(d) Condensed frame #300.

(c) Original frame #1,690.

Figure 9: Result of ribbon carving with $\phi_{max} = 2$ for the *Sidewalk* sequence: (a-c) three original frames; (d) frame from the condensed sequence comprising objects from different original frames (times).