Compact Video Synopsis via Global Spatiotemporal Optimization

Yongwei Nie, Chunxia Xiao, *Member, IEEE,* Hanqiu Sun, *Member, IEEE,* Ping Li

Abstract—Video synopsis aims at providing condensed representations of video datasets that can be easily captured from digital cameras nowadays, especially for daily surveillance videos. Previous work in video synopsis usually moves active objects along the time axis, which inevitably causes collisions among the moving objects if compressed much. In this paper, we propose a novel approach for compact video synopsis using a unified spatiotemporal optimization. Our approach globally shifts moving objects in both spatial and temporal domains, which shifting objects temporally to reduce the length of the video and shifting colliding objects spatially to avoid visible collision artifacts. Furthermore, using a multi-level patch relocation method, the moving space of the original video is expanded into a compact background based on environmental content to fit with the shifted objects. The shifted objects are finally composited with the expanded moving space to obtain the high-quality video synopsis, which is more condensed while remaining free of collision artifacts. Our experimental results have shown that the compact video synopsis we produced can be browsed quickly, preserves relative spatiotemporal relationships, and avoids motion collisions.

Index Terms—Video synopsis, surveillance, optimization, patch relocation.

1 Introduction

With the rapid development of the digital-media industry, the huge video datasets captured by various resources such as digital cameras, webcams, cellular phones, PDAs, and surveillance cameras, are growing at an explosive speed. It is time consuming to review entire, lengthy videos, such as those captured by surveillance cameras, to find interesting objects, since most surveillance videos contain only a limited number of important events. Thus, end users often prefer briefer, condensed representations of long video sequences, fast-forwarding to important content and dynamic objects in surveillance videos. This is generally referred as video synopsis or abstraction [1], [2], [3], [4], [5]. In addition, it is memory intensive to store and transfer the entire captured videos while retaining less important objects; thus, video synopsis can effectively reduce memory storage for large video datasets. Furthermore, video synopsis techniques can be widely used as practical video editing toolkits, to abstract videos captured in video games, video crowd animation, and video conferences. Currently, video synopsis is an active research topic in the computer graphics and computer vision communities.

To date, there is no unified standard for measuring if an output abstracted video is a good representation of an input video, since whether a synopsis is good or bad is highly subjective and depends on the application. In general, we outline three main objectives for our video synopsis: the spatiotemporal redundancies of the input video should be reduced as much as possible; the chronological consistency of important events should be kept; the visual artifacts, such as flickering or moving object collisions, should be avoided in the final synopsis. Efficient browsing of the condensed video synopsis is also important, especially for huge video datasets such as surveillance videos.

Many approaches have been proposed for condensing the volume of videos. Most of them are based on video frames, where image frames are treated as the basic building blocks that cannot be decomposed. In video abstraction methods [1], [3], [6], [7], either key frames were selected according to some importance criteria or video clips with lower interest or activity were skipped. One typical approach [8] used a time-lapse method to generate a summary of a very slow process, such as the growth of a flower over an entire day. These methods miss fast activities occurring in skipped frames. In order to reduce less important spaces in the video volume, Kang et al. [2] extracted informative space-time video portions, and then montaged these portions together. However, in this way, visible seams may appear at the boundaries between different portions. More recently, Pritch et al. [4], [9], [10] proposed an object-based video synopsis approach for surveillance videos, in which objects are shifted along the time axis. Although this approach is capable of eliminating temporal free space, it produces unpleasant collisions between moving objects in the condensed video, especially for videos with narrow motion space.

In most site-seeing surveillance videos, we often observe large amounts of free space in the scene backgrounds, where no active objects move at all. Previous

- *Yongwei Nie and Chunxia Xiao are with the Computer School of Wuhan University, Wuhan, China, 430072.*
  E-mail: nieyongwei@gmail.com, cxxiao@whu.edu.cn
- *Hanqiu Sun and Ping Li are with the Department of Computer Science & Engineering, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong.*
  E-mail: hanqiu@cse.cuhk.edu.hk, pli@cse.cuhk.edu.hk

work [4] extracted active objects from the input video and moved them compactly along the time axis, which can certainly condense videos in the temporal space. However, when there are many objects moving in the same orbit or in opposite directions, unavoidable collisions among them occur, which inevitably produces unpleasant artifacts, such as objects moving across each other. The key observation in this paper is that the free space in the video background can be utilized when condensing the video. Based on this observation, we construct larger virtual motion spaces (i.e. multi-paths) for moving objects based on the environmental context, which effectively alleviates the moving-object collisions produced in previous video synopsis methods. Thus, the condensed video looks more realistic even in shorter periods.

Our key idea in this work is to globally shift the active objects in the spatiotemporal video volume, and then synthesize a compact background constrained by the optimized object trajectories to fit the shifted objects. We propose a global spatiotemporal optimization framework that shifts active objects in the spatiotemporal space. The optimization is composed of two cost terms: the data term is used to preserve activities as much as possible and prevent objects' new trajectories from deviating far from their original ones, and the smoothness term is used to avoid object collisions and keep the spatiotemporal consistencies and relations of objects as much as possible. We develop a multilevel patch relocation method to synthesize the compact background, which further expands the movement space of shifted objects based on the environmental context, so that visual artifacts such as cars running on grassland or mutual crossings can be eliminated. Finally, we seamlessly fuse the shifted objects into the compact background to produce the final video synopsis.

Our framework presents a compact video synopsis technique via global spatiotemporal optimization. Using the synthesized compact background and simple user interactions, our approach can produce more condensed video scenes with crowded but non-colliding objects from input videos containing sparse moving objects, and can be widely applied in video summarization, video games, crowd animation design, and interactive media production. Our work makes the following two main contributions:

- We propose a novel approach for compact video synopsis in the spatiotemporal domain, which highly condenses the activity information for surveillance videos, while avoiding visual collision artifacts between moving objects;
- We introduce a synthesized compact background which provides a larger virtual motion space for shifted objects using a multilevel patch relocation method in Markov Random Field (MRF) networks.

2 Related Work

Here, we review the most related work in video synopsis, which can be roughly classified as frame-based video abstraction, object-based video synopsis, and synthesis-based video summarization. Previous work in patch-based image editing related to our background-synthesis work is also outlined.

Frame-based abstraction There are two basic forms of video abstracts: keyframes and video skims. They are both based on frame selection, where the frames are essential building blocks that can't be decomposed. In the keyframe-based abstraction methods, a set of salient frames are extracted from the source video. Uniform sampling is the simplest method for keyframe generation, but it may select some frames that aren't as "key" as desired. Methods for finding importance criteria to guide the selection process were proposed in [1], [6], [11], [12]. The fast-forward methods [7], [13] are another kind of keyframe-based method, which select keyframes uniformly or adaptively, but they do not preserve time coherence and may result in unrealistic views. High-level content analysis is also taken into account in keyframe selection processing [14], [15]. Alternatively, video skims [16], [17] consist of a collection of video segments extracted from the source video. These segments are then joined by a cut or a gradual effect. Although the above two kinds of methods work well in most situations, they suffer from loss of fast activities occurring in the skipped frames, and retention of large empty spaces in natural scenes from the source videos.

Object-based synopsis Kang et al. [2] explicitly extracted informative space-time video portions that can be moved and stitched using first-fit and graph-cut optimizations which maximizes the amount of visual information. As optimal boundaries between portions are found, visually unpleasant seams usually appear between portions that don't match. Rav-Acha et al. [9] proposed an object-based approach for video synopsis that is similar to that of Kang et al. [2]. Both of these methods change the chronological order of objects, and show several actions at the same time. The difference is that the latter [9] only moves objects along the time axis. Pritch et al. [10] extended the latter work [9] to process always-on videos captured by surveillance cameras or webcams. The more complete work is presented in [4], in which they first extracted interesting objects (tubes), then moved them along the time axis while preserving activities and local chronological orderings. It can handle always-on videos and take illumination-varying backgrounds into account. However, since it processes moving objects only in the temporal domain, this method cannot fully utilize the video space in the spatial domain and may produce collision artifacts when the synopsis is compressed much or contains more moving objects.

Synthesis-based summarization The synthesis-based video summarization approach was proposed in [5], which defined a bidirectional similarity measure to de-

cide how visually similar a summary was to the underlying source video. A good summary contains as much visual information from the source as possible, and introduces as few new visual artifacts as possible. The best-matching patches in the source and summary are found and then go through a weighted average process to obtain the final summary. Since the nearest neighboring patches need to be computed and stored, the algorithm is time and memory intensive. Barnes et al. [18] and Xiao et al. [19] proposed fast patch match methods which can accelerate the nearest neighbor search to a certain extent, but the synthesis-based approach is still not scalable for long videos.

Patch-based image editing Like pixels, patches can also be used as the basis for analyzing and editing images. Features are either extracted from patches for further analysis, or the patch is directly operated on for image editing. It is often more effective to work on patches than on pixels. Kwatra et al. [20] proposed a patch-based texture optimization framework which improves texture quality compared to pixel-based methods [21]. Xiao et al. [22] obtained effective upsampling results by combining patch-based texture synthesis and joint bilateral filter. Image and video completion methods [23], [24], [25] also work on the patch level. Cho et al. proposed a patch transformation method [26] that provides image editing tools for image reorganization, object removal, image retargeting, etc. More recently, Cho et al. [27] edited images by working on overlapped patches, and a patch jittering post-processing step was proposed to improve the quality of the edited image. In our work, we improve the method of [27] to synthesize the compact background.

3 Overview

Fig. 1 illustrates the main motivation of our proposed approach. Given an input video (Fig. 1a), visual motion collisions occur when shifting objects along the temporal axis (Fig. 1b). Using the spatial free space observed in most site-seeing surveillance videos, we attempt to shift active objects in both the spatial and temporal domains (Fig. 1c). In this fashion, we not only condense temporal free space, but also make full use of spatial free space for movements, which can effectively avoid visual artifacts, such as moving-object collisions, producing high-quality video synopsis results with scene-path context.

Our video synopsis framework is composed of four main stages. In the first stage, we analyze the input video and explicitly extract the background and moving objects out of the video. In the second stage, we shift active objects in the spatiotemporal video volume by using the global spatiotemporal optimization, which computes new positions in the synopsis for clustered objects. In the third stage, using a multilevel patch relocation method, we synthesize a compact background with the scene-path context to fit the clustered objects. Lastly, we seamlessly fuse objects into the synthesized compact background using a gradient-domain editing tool.

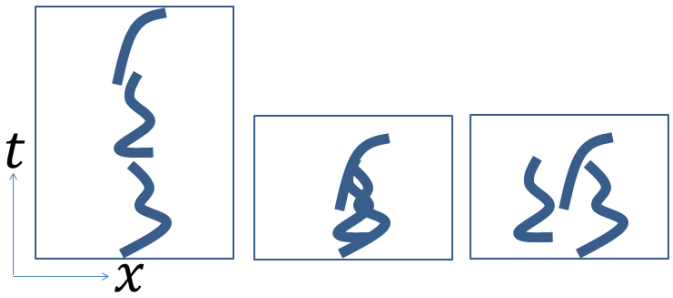


Fig. 1. (a) Input video. (b) Temporal shift of objects, reducing video length by eliminating temporal free space but producing collisions among the moving objects. (c) Our global shift optimization avoids motion collisions between objects

Fig. 2 shows the main advantages of our proposed approach. Fig. 2b shows that shifting objects only in the temporal domain leads to heavy object collisions (the circled cars, for example). Our global spatiotemporal optimization effectively eliminates these collisions (Fig. 2c). Since the original video background (Fig. 2d) may not be compatible with the shifted objects (cars running out of the road in Fig. 2e), we compute a larger virtual motion space by synthesizing a compact background to remove these inconsistencies (Fig. 2f). Finally, the shifted objects are seamlessly composited into the compact background, producing an appealing video synopsis (Fig. 2g).

4 Compact Video Synopsis

In this section, we first analyze the input video to extract the background and moving objects. Then, we describe our global spatiotemporal optimization in detail. Finally, we present the compact background synthesis using the scene-path context, and the method to seamlessly fuse the clustered moving objects into the compact background.

4.1 Video Analysis

We use a space-time volume $I(x, y, t)$ to represent an input video I, where (x, y) is the spatial coordinate of a pixel in frame t, satisfying $(1 \leqslant x \leqslant W)$, $(1 \leqslant y \leqslant H)$ and $(1 \leqslant t \leqslant N)$. With the input video, we first extract background B and interesting objects s. Typically, interesting objects are simply defined as moving objects, such as running cars or walking people. Sometimes, exceptions may be noted: not all moving objects are interesting, and not all static objects are less important. To handle such exceptions, more sophisticated techniques in pattern recognition [28] can be incorporated.

Before extracting active objects and synthesizing the compact background, we first extract the background of the input video. For most site-seeing surveillance

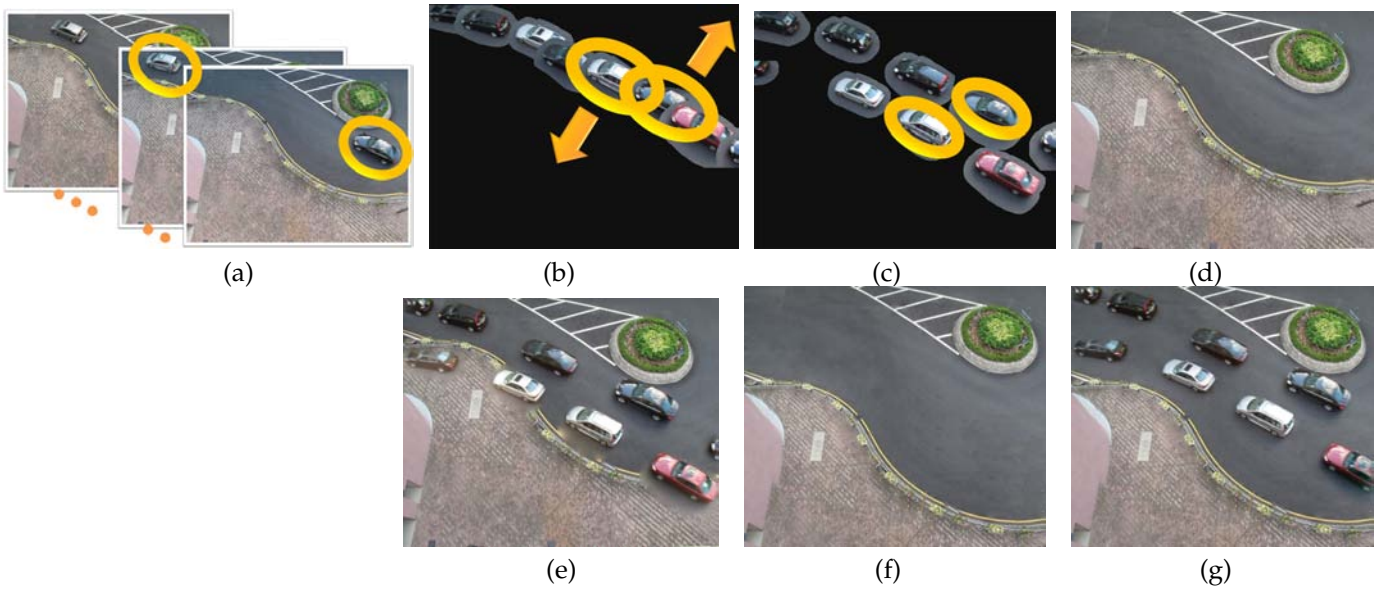


Fig. 2. (a) Input video (total running time: 3 mins) with two interesting objects marked. (b) Temporal shifting [4] produces collisions if compressed too much (total running time: 23s) (two marked cars as example). (c) Objects shifted globally to avoid such artifacts. (d) The original extracted background. (e) Shifted objects not compatible with the original background. (f) The synthesized compact background. (g) Shifted objects composited into the compact background, producing an even shorter synopsis (total running time: 15s) without visual artifacts.

videos, video frames change due to the entering and exiting of moving objects and due to varying illumination. The background is usually static in daily periods. Based on this observation, we use a temporal median operator over a short period to extract the corresponding background. The median value of pixels with the same location (x, y) in a short video clip makes up the background value of the location. In our experiments, we compute a background image every minute (1800 frames for a 30 Fps video), which can alleviate the effects of varying illumination. For more complex situations, such as moving objects covering pixels over a long duration, we recommend using a shorter temporal window background estimation method [29]. With the background extracted, the activity measure $\Theta(x, y, t)$ for pixel $I(x, y, t)$ is defined as the difference between the pixel and its corresponding background value:

$$\Theta(x, y, t) = \|I(x, y, t) - B(x, y)\|, \quad (1)$$

where $B(x, y)$ is the pixel value at coordinate (x, y) in background B.

Next, we explicitly detect and extract interesting objects from the input video. As the objects will be later composited back into the background using Poisson Video Editing [30], precise object extraction is not necessary. In our system, we first subtract the extracted background from the original frames. Then, we construct a mask of all foreground pixels with greater absolute differences than a threshold value (10 out of 255 is used). We apply a 2D morphological dilation on all the mask frames to obtain larger moving masks. The morphological mask is a circle with a radius of 3 pixels, and we iterate the dilation three times. If needed, the background cut by Sun et al. [31] can be used to precisely segment foreground objects, and is more robust but computationally more expensive. We connect all object masks together across frames to construct a 3D tube of moving object. However, when the objects move fast or overlap with each other, we perform real-time tracking [32] to construct object tubes. Fig. 3a shows one frame of input video, Fig. 3b shows the extracted background and Fig. 3c shows the extracted moving object.

4.2 Global Spatiotemporal Optimization

The spatiotemporal operator $V_s = (s_x, s_y, s_t)$ of interesting object s, where s_x and s_y are spatial offsets and s_t is temporal offset, shifts the object to position $\tilde{s}$ in the synopsis by $\tilde{s} = s + V_s$. Most objects are shifted forward along the timeline, so that the length of synopsis O, denoted by M, is much shorter than that of the input video. We compute the optimal new positions for the interesting objects by using a global spatiotemporal optimization which minimizes the following Gibbs function:

$$E(V_s) = E_{data}(V_s) + E_{smooth}(V_s). \quad (2)$$

The data term is defined on single objects s_i:

$$E_{data}(V_s) = \sum_{s_i} (E_a(\tilde{s}_i) + \gamma E_d(\tilde{s}_i)), \quad (3)$$

where E_a encodes activity cost, and E_d is spatial distance cost. The smoothness term is defined on pairs of objects s_i and s_j:

(a) (b) (c)

Fig. 3. (a) One frame of input video. (b) Extracted background. (c) Extracted object.

$$E_{smooth}(V_s) = \sum_{s_i,s_j \in S} (\alpha E_{st}(\tilde{s}_i, \tilde{s}_j) + \beta E_c(\tilde{s}_i, \tilde{s}_j)) \cdot \delta(\tilde{s}_i, \tilde{s}_j), \tag{4}$$

where,

$$\delta(\tilde{s}_i, \tilde{s}_j) = \begin{cases} 1, & if \quad \tilde{s}_i \neq \tilde{s}_j, \\ 0, & otherwise. \end{cases} \tag{5}$$

The smoothness term is the sum of weighted collision cost E_c and spatiotemporal consistency cost E_{st}. α, β and γ are weights that can be tuned by users, and their default values are 2, 5, and 0.5 in our experiments.

E_a favors a synopsis with maximum activities by penalizing objects moving outside of the synopsis. It is zero if an object stays in the synopsis as a whole. Otherwise, it is sum of the activity values of pixels outside the synopsis.

$$E_a(\tilde{s}_i) = \sum_{(x,y,t) \in \tilde{s}_i \setminus synopsis} \Theta_{\tilde{s}_i}(x,y,t), \tag{6}$$

where $(x,y,t) \in \tilde{s}_i \setminus synopsis$ represents pixels belonging to $\tilde{s}_i$ but not the synopsis, and $\Theta_{\tilde{s}_i}(x,y,t)$ is the pixel activity value defined in Eq. (1).

E_d prevents objects from being shifted far away from their original trajectories in the spatial domain; otherwise, objects may scatter anywhere in the synopsis and serious inconsistencies among them would occur. We define it as object spatial distance:

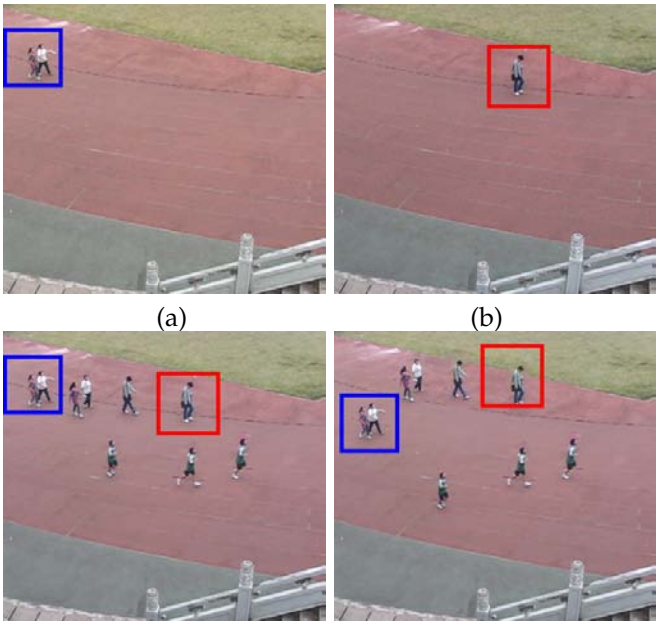
$$E_d(\tilde{s}_i) = n_{s_i} \|(\tilde{s}_{i,x}, \tilde{s}_{i,y})\|^2, \tag{7}$$

where n_{s_i} is the number of pixels of object s_i (this balances the spatial distance cost against other costs that use all pixels of an object, and not just one). In Fig. 4, we illustrate the comparative results with and without using spatial distance cost E_d, which shows that shifted objects do not drift far away from their original positions when using the cost.

E_c encodes the collision cost of all pairs of objects. For the overlapped part of two shifted objects $\tilde{s}_i$ and $\tilde{s}_j$, we define the collision cost as the sum of the products between pixel activities of two objects:

$$E_c(\tilde{s}_i, \tilde{s}_j) = \sum_{(x,y,t) \in \tilde{s}_i \cap \tilde{s}_j} \Theta_{\tilde{s}_i}(x,y,t) \cdot \Theta_{\tilde{s}_j}(x,y,t). \tag{8}$$

We see that the collision cost is zero if two objects don't collide with each other. We shift the objects in both the spatial and temporal domains, which expands the motion space for objects and effectively reduces collision artifacts.



(a) (b) (c) (d)

Fig. 4. Comparative results with and without using E_d. (a) and (b) are two frames from the input video. (c) Using E_d, the two marked objects are not shifted far away from their original trajectories and are perfectly fused into the background. (d) Without E_d, the marked objects are shifted away from their original trajectories.

E_{st} preserves the spatiotemporal relations of objects: (1) Chronological order should be kept, i.e., objects behind other objects should not appear in front of them. The cost of reversing the chronological order of two objects is the sum of their activities; when the chronological order is kept, the cost is zero. (2) The spatial relative locations of two objects should be preserved if they are neighboring each other. The cost of breaking such relationships is defined as the difference of their spatial offsets, which is then weighted by the original distance of the two objects. Let $t^f_{s_i}$ and $t^f_{s_j}$ be the first frames of two objects s_i, s_j. We compute a variable $u = (t^f_{s_i} - t^f_{s_j}) \cdot (t^f_{\tilde{s}_i} - t^f_{\tilde{s}_j})$. Then we determine whether the chronological order of the two objects is broken or not by:

$$\tau(u) = \begin{cases} 0, & if \quad u \geq 0, \\ 1, & else. \end{cases} \tag{9}$$

We determine whether two input objects are neighboring each other by checking if they share common frames. If so, we compute the nearest distances of the two objects for all common frames, from which we find the minimum distance $d(s_i, s_j)$; otherwise, $d(s_i, s_j)$ is set as ∞. Let $\chi_{s_i} = \sum_{(x,y,t) \in s_i} \Theta_{s_i}(x,y,t)$ be the object's activity. Then, we define the spatiotemporal consistency

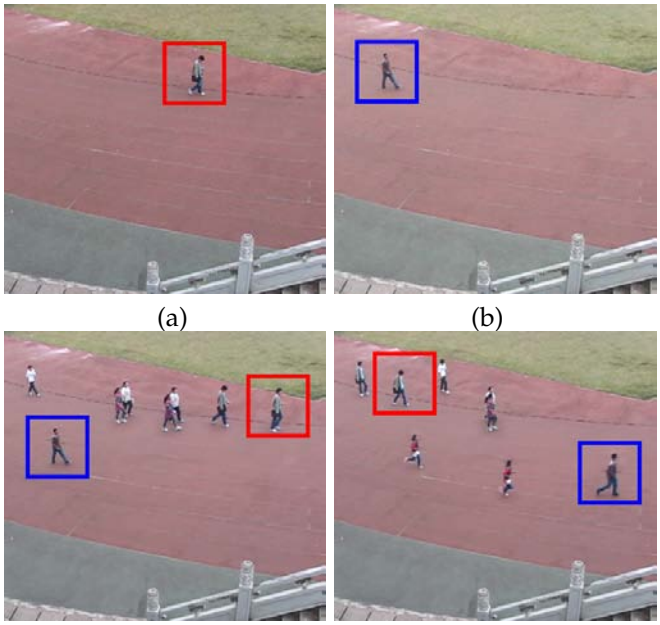


Fig. 5. Comparative results with and without using E_{st}. (a) The input 1200th frame. (b) The input 10280th frame. (c) Using E_{st}, the two objects keep their chronological order in the synopsis. (d) Without E_{st}, the blue object walks in front of the red one, which is not the case.

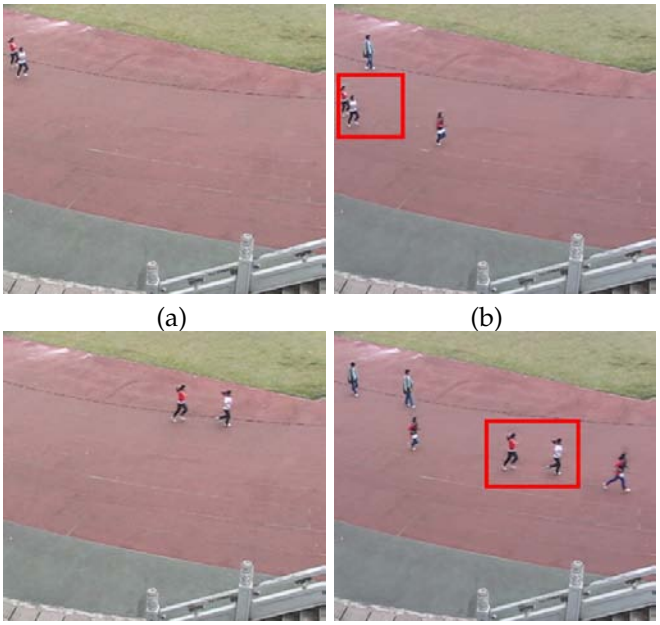


Fig. 6. The relative spatial relations of objects are kept in the synopsis. (a) The 2400th frame from the input video. (b) The 100th frame from the synopsis (corresponding to (a)). (c) The 2480th frame from the input video. (d) The 180th frame from the synopsis (corresponding to (c)).

cost as:

$$E_{st}(\tilde{s}_i, \tilde{s}_j) = \tau(u) \cdot (\chi_{s_i} + \chi_{s_j}) + (n_{s_i} + n_{s_j}) \cdot exp(-d(s_i, s_j)/\sigma_{st}) \cdot \|(\tilde{s}_{i,x}, \tilde{s}_{i,y}) - (\tilde{s}_{j,x}, \tilde{s}_{j,y})\|^2, \quad (10)$$

where σ_{st} determines the extent of how close the two objects are, and is set as 40 in our experiments. Fig. 5 shows that using our spatiotemporal consistency cost E_{st}, the chronological order among objects is preserved. We show in Fig. 6 that the spatial relations among objects are preserved.

$\delta_(\tilde{s}_i, \tilde{s}_j)$ makes Eq. (2) regular according to the theoretic results of [33], which guarantees the equation is graph-representable. We use the Graph Cuts technique to minimize Eq. (2).

Energy Minimization We use alpha-beta swap Graph Cuts method [34] to minimize the global spatiotemporal optimization given in Eq. (2). First, we construct a graph for the synopsis where each node N_i in the graph corresponds to an input object s_i, and the edge $E_{i,j}$ connects nodes N_i and N_j. The cost of node N_i is $E_a(\tilde{s}_i)+\gamma E_d(\tilde{s}_i)$ when assigning label $\tilde{s}_i$ to it, and the cost of edge $E_{i,j}$ is $(\alpha E_{st}(\tilde{s}_i, \tilde{s}_j) + \beta E_c(\tilde{s}_i, \tilde{s}_j)) \cdot \delta(\tilde{s}_i, \tilde{s}_j)$ when assigning labels $\tilde{s}_i$ and $\tilde{s}_j$ to nodes N_i and N_j, respectively. With the constructed graph, the minimum cut of the graph minimizes Eq. (2).

For each node N_i, the number of its possible labels is the number of pixels in the synopsis $W \times H \times M$. The label number is usually too large for efficient computation. For instance, there are about $10^9(640\times480\times60\times30)$ labels for a 60-second synopsis with a resolution of 640×480 and FPS of 30. To speed up the computation, we sample the synopsis pixels in the spatiotemporal space using grids of size $\eta = 20$ pixels in our experiments, which results in a much smaller label set $W \times H \times M/\eta^3$. For the same-length synopsis, the label number is 69120$((60 \times 30 \times 640 \times 480/20^3))$ now, which is still somewhat time-consuming.

To further reduce computational cost, we approximate the global optimization by splitting it into two consecutive optimizations, which degrades the label space from 3D to 1D and 2D. We first constrain the spatial offsets s_x and s_y to be zero and only shift objects along the temporal axis using temporal optimization. Then, we constrain the temporal offset s_t to be zero and shift objects into the spatial free space using spatial optimization. Now, for the 60-second synopsis, the spatial shift optimization has only 768$(640 \times 480/20^2)$ labels. However, for the temporal shift optimization, the procedure may become trapped in a local minimum. For example, one special case may occur: all the objects are inside the synopsis, and are shifted into the same label. Both the data term and smoothness term are zero in this case, but it is not the desired result. To avoid the local minima, we expand the temporal label space from M/η to $n \cdot M/\eta$, where n is the number of nodes. In the graph construction, we set the valid label range $[i \cdot M/\eta, (i+1) \cdot M/\eta)$ (otherwise invalid range) for each node $N_i(0 \leqslant i \leqslant n)$. For invalid labels, the costs of node N_i and edge $E_{i,j}$ are set to ∞. After optimization, for an object s_i with label $\tilde{s}_i$, its temporal position is $\tilde{s}_i$ mod M/η. For the 60-

second synopsis with 15 moving objects, the temporal optimization has only 1350 $(15\times60\times30/20)$ labels. Using this approximation method, we greatly speed up global spatiotemporal optimization in our system.

4.3 Iterative Object Shifting

(a) (b) (c)

Fig. 7. Iterative object shifts. (a) Without E_{SC}, the shifted objects may be distributed freely. (b) With E_{SC}, by setting $K = 2$, the shifted objects are clustered into 2 groups, and each of them moves with the same vector. (c) The iterative shifting result produced by setting $K = 3$.

In the global optimization, the spatiotemporal consistency cost E_{st} keeps the spatial relations of any two neighboring objects. For the example in Fig. 7a, the collisions among objects are eliminated and the relative positions among pairs of neighboring objects are preserved, even though the objects are distributed somewhat unordered in the overall view. To further refine the synopsis results, we try to keep the relative positions among multiple objects rather than between pairs of neighboring objects, using an iterative object shifting scheme.

The basic idea is that objects are clustered into several groups, and objects in a group are shifted with the same vector in the spatial domain. After energy function Eq. (2) is minimized, we divide the shifted objects into K groups based on the calculated spatial offsets $(\tilde{s}_{i,x}, \tilde{s}_{i,y})$ using the K-Means method. The objects with similar shift vectors are assigned to the same group. For each of the groups, we compute its centra $(c_{k,x}, c_{k,y})$, where $k \in [1, K]$. Then, we define the spatial consistency cost E_{SC} for a shifted object $\tilde{s}_i$ as the difference between its offset and the centra of the cluster it belongs to:

$$E_{SC}(\tilde{s}_i) = n_{s_i}\|(\tilde{s}_{i,x}, \tilde{s}_{i,y}) - (c_{k,x}, c_{k,y})\|^2, \qquad (11)$$

where $\tilde{s}_i$ belongs to the $k-th$ group. Finally, the data term $E_{data}(V_s)$ in Eq. (3) is redefined by adding the spatial consistency cost:

$$E_{data}(V_s) = \sum_{s_i}(E_a(\tilde{s}_i) + \gamma E_d(\tilde{s}_i) + \epsilon E_{SC}(\tilde{s}_i)), \qquad (12)$$

where the weight ϵ is set as 0.5 in our experiments.

With this new data term, we minimize Eq. (2) iteratively to obtain better global shifting of interesting objects. The iterative scheme works as follows: first, it divides objects into groups according to the spatial shifting vectors computed in the previous iteration, then it computes group centra and minimizes Eq. (2) with the data term shown in Eq. (12) to output new object shifting vectors. This procedure goes on about 4 or 5 times until the positions of shifting objects converges. By setting K as 2 or 3, Fig. 7b and 7c show that the cars are grouped into two or three queues using our iterative object shifting.

4.4 Compact Background Synthesis

Our global spatiotemporal optimization expands the motion space of objects, which leads to the problem of presenting a compatible background for shifted objects. This problem will not occur in scenes that are full of moving space, for example in Fig. 4, 5, 6, where people can run on every athletic track. However, in other scenes, there may be not enough moving space for the shifted objects, for example in Fig. 2e and Fig. 7, with cars running out of road. The task in this section is to synthesize a compact background that expands the moving space, constrained by the shifted objects and by user interactions. The synthesized compact background must fit all shifted objects to avoid unrealistic artifacts. We use a patch relocation scheme to achieve this goal. We also develop a multilevel relocation method to further improve the quality of this synthesis.

Patch Relocation Inspired by patch transformation [27], we relocate existing image patches of extracted background to generate a compact background. We divide the background into grids (grid size η) and construct a MRF on the grids, where each grid is viewed as a node in the MRF. The problem is how to assign each MRF node an index x_i of patch (here the grid is considered as a patch). In a good relocation of patches, adjacent patches should fit each other and the relocation should be subject to the constraints of shifted objects. With the MRF framework, the above requirements are formulated as a joint probability:

$$P(X) = \prod_i \Phi_i(x_i) \prod_{i,j\in N(i)} \Psi_{i,j}(x_i, x_j), \qquad (13)$$

where $\Psi_{i,j}(x_i, x_j)$ is the compatibility of two patches at node i and j; $\Phi_i(x_i)$ is the local evidence term used to represent the constraints. To maximize this joint probability using loopy belief propagation [35], Eq. (13) is factorized into:

$$P(X) = \prod_i \prod_{j\in N(i)} p(y_i|x_i)p_{i,j}(x_j|x_i)p(x_i), \qquad (14)$$

where for node i, $N(i)$ are the indices of its four neighboring nodes; x_i is the index of the patch assigned to node i; y_i is its original patch index at location i; $p_{i,j}(x_j|x_i)$ is the normalized compatibility between patches x_i and x_j with respect to the relative locations between nodes i and j, which is defined as:

$$p_{i,j}(x_j|x_i) = \frac{\Psi_{i,j}(x_i, x_j)}{\sum_k \Psi_{k,j}(x_k, x_j)}. \qquad (15)$$

As for local evidence, $p(y_i|x_i) = \Phi_i(x_i)$. Finally, $p(x_i)$ is modeled as a uniform distribution. The compatibility term $\Psi_{i,j}(x_i, x_j)$ is formulated as:

$$\Psi_{i,j}(k,l) = exp(\frac{-E_{seam}(k,l)}{\sigma_\Psi^2}), \quad (16)$$

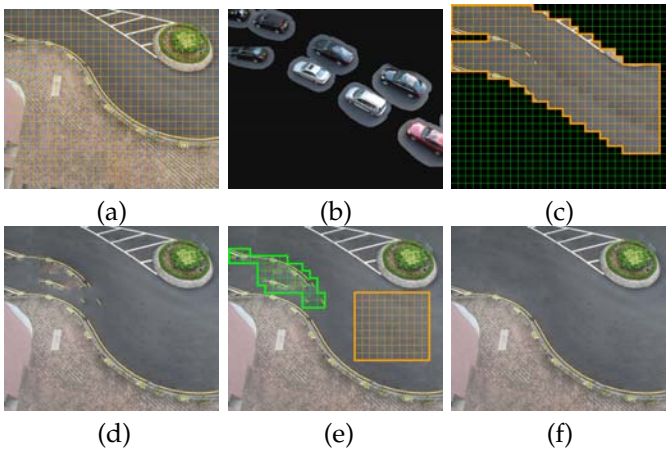


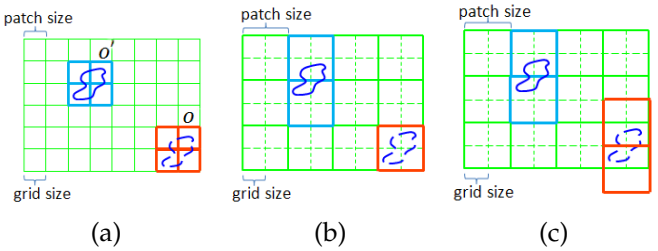
Fig. 8. Patch relocation. (a) The extracted background image is divided into grids, where the orange grids are patches. (b) One frame of shifted objects. (c) The initial compact background (the orange regions) is constructed based on the shifted objects, using them as constraints for patch relocation. (d) Patch relocation automatically produces a compact background subject to these constraints. (e) Local refinement on the manually-selected green region removes artifacts, using orange regions as source patches. (f) The final compact background in the video synopsis.

where $E_{seam}(k,l)$ is the minimum energy among continuous seams between two patches k and l, calculated by dynamic programming [36]. σ_Ψ is fixed as 0.2 after cross validation. With the local evidence term, we determine how likely it is for a patch to be assigned to a node. For the constrained nodes, if fixing patch k for node i, we set $p(y_i|x_i = k) = 1$, and $p(y_i|x_i = l) = 0$ for any $l \neq k$. In our method, the constrained nodes are the ones occupied by shifted objects $\tilde{s}_i$ in the synopsis. The corresponding patches occupied by object s_i in the input video are set for those constrained nodes. For the unconstrained nodes, $p(y_i|x_i)$ is computed as the difference between the assigned patch l and the original one y_i:

$$p(y_i|x_i = l) \propto exp(-\frac{(m(y_i) - m(l))^2}{\sigma_{evid}^2}), \quad (17)$$

where $m(\cdot)$ is the mean color of argument and σ_{evid} is fixed as 0.4.

Fig. 8 gives an example of patch relocation. We use Loopy Belief Propagation to maximize Eq. (14) (refer to [27] for more detail). The global patch relocation may sometimes retain artifacts as shown in Fig. 8d. We use a local refinement technique to remove the artifacts. As illustrated in Fig. 8e, the user explicitly indicates the artifact region and it is then divided into grids (the green ones) to construct a local MRF. Using the source patches (the orange ones) selected by the user input, the local patch relocation is performed on these nodes and patches. Fig. 8f shows the final compact background synthesized after the local relocation step.



(a) (b) (c)

Fig. 9. (a) The object O is shifted to the position of O'. When the patch size equals the grid size, the red patches match the four blue patches. (b) With patch size larger than grid size, the two blue patches of the shifted object can't find matched patches containing the original object. (c) We sample the patches every grid size rather than every patch size, to supply more sufficient patch samples for good matching.

Multilevel Patch Relocation The patch relocation is modeled on low-level vision. When synthesizing a complex compact background with strong structure or the background is densely sampled in grids, a single patch relocation process may fall into local minima easily (Fig. 10c). This is due to the fact that a small patch itself can't contain much image structure. For complex scenes, we prefer a larger patch size to preserve important structures; however, serious visual seams among big patches occur (Fig. 10e). To flexibly utilize different patch sizes, we develop multilevel patch relocation in our synopsis system. We first determine the compact appearance at coarser levels (larger patch sizes), and then refine the detail at finer levels. In our experiments, the patch size in the coarsest and finest levels varies from 80 to 20 pixels, respectively. The number of MRF nodes at the finer level is 4 times than that of the coarser level. In two consecutive levels, we use the result of the coarser level to initialize the BP iterative procedure of the finer one.

As we shift objects grid by grid, mismatching problems may occur when the patch size is not equal to the grid size. In Fig. 9, we show a correct case in (a) and a problem case in (b). In Fig. 9c, we give our solution to (b). In Fig 9a, a grid is a MRF node and we sample patches from grids. If the grid and patch (and node) share a common size, the nodes of shifted object O' (blue ones) can find the corresponding patches of object O (red ones). In Fig. 9b, the patch (node) size is double, and each patch consists of four grids now. The mismatching problem happens because the shifted object O' occupies two nodes but the original object only occupies one patch. In Fig. 9c, we sample patches by an interval of a grid cell in both x and y directions. Then, we can find

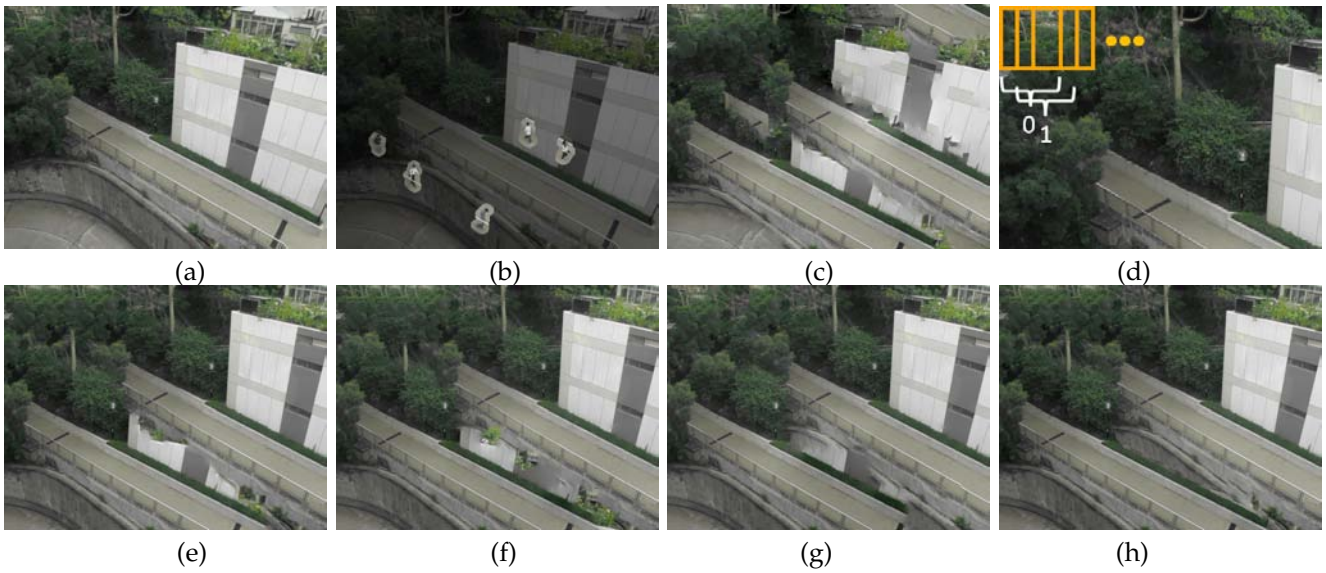


Fig. 10. Multilevel patch relocation. Three levels are used, node sizes are 80, 40, and 20. (a) Extracted background. (b) The shifted objects. (c) Using single-level patch relocation with patch size 20, the wall of the building becomes trapped in a local minimum. (d) Sampling patches for the coarser levels every 20 pixels rather than 80 (or 40). (e) Compact background on 2-th level (patch size is 80). (f) Compact background on 1-th level (patch size is 40). (g) Compact background on 0-th level (patch size is 20). (h) Compact background after local refinement.

two matched patches (red ones) for both blue nodes. We use this method for the coarser levels to eliminate the mismatching problem. Fig. 10e, 10f and 10g give the results of our multilevel patch relocation from the coarsest to the finest levels, validating the effectiveness of our method.

Finally, using the generated shifted objects and the compact background, we seamlessly compose them to produce the final video synopsis with more compact views and a shorter synopsis length. In our experiments, we used Poisson cloning [30] for video composition, which worked well for our task.

5 Results and Discussion

We have developed our compact video synopsis system in C++, and run it on an Intel Core 2 Duo 2.1GHz CPU computer with 2 GB RAM. We have tested our approach on several surveillance video examples (frame rate of 30 Fps), and compared our results with previous methods. In Table 1, we give relevant experimental information and parameters for each example: the length of the input video (Input Len.), the number of interesting objects (Obj. Num.), the length of the synopsis (Synopsis Len.), the number of groups K in the iterative object shifting method, the number of labels L_t and L_s in the temporal and spatial optimizations, and the time T_t and T_s consumed by the two optimizations, respectively. We show whether the multilevel patch relocation (MPR) method and/or the local refinement (LR) scheme are used. We give the running times of [4] in the last column.

We can see from the table that all the input videos are longer than 5 minutes (30 minutes maximum) and that the synopsis results are much shorter (15 seconds minimum). Our global spatiotemporal optimizations contain few enough labels, and most of them can be finished within 22 seconds. In the compact background synthesis, computing compatibility $\Psi_{i,j}(x_i, x_j)$ between two patches consumes most of the time. The total time used for compact background synthesis is less than 29 seconds. Since our system includes both spatiotemporal optimization and compact background synthesis, we consume more time than [4] for more compact and shorter video synopsis results.

In Figs. 11, 12, 13, 14 and 15, we give video synopsis results for site-seeing surveillance videos and compare our work with [4]. The previous work is powerful in compressing time-scale redundancies but may produce collision artifacts if compressed much. The video synopsis examples we tested showed that our approach produces more compact and shorter synopsis results without collision artifacts. The video synopsis demos are included in the online video submission.

In Fig. 11, the input video (10 minutes) shows a bridge scene where many people cross randomly. Shifting them only along the time axis as in [4], people collide with each other easily, especially when people walk in opposite directions, since the movement space is very narrow. The top row shows four frames with collisions from the synopsis (2 minutes) of [4]. The collision artifacts (i.e. moving objects colliding on the bridge) make the synopsis flicker and inconsistent. The bottom row shows the results of our spatiotemporal synopsis. Our result is shorter in length (1 minute), and there are no collisions. We achieve this by expanding the movement space into free spatial domain in an environmental-path context

TABLE 1
Experimental parameters and run-time information in video synopsis.

Examples	Input Len.	Obj. Num.	Synopsis Len.	K	L_t	L_s	T_t	T_s	MPR	LR	Run-time [4]
Fig. 11	10 minutes	16	1 minutes	3	1440	768	22 s	29 s	Yes	Yes	12 s
Fig. 12	3 minutes	18	15 s	2	396	720	15 s	25 s	No	Yes	6 s
Fig. 13	30 minutes	21	1 minutes	3	1890	672	28 s	26 s	Yes	Yes	22 s
Fig. 14	5 minutes	6	24 s	3	216	351	6 s	20 s	Yes	Yes	3 s
Fig. 15	10 minutes	16	40 s	2	960	768	19 s	10 s	Yes	Yes	10 s

Fig. 11. Input video (10 minutes). Top: Video synopsis result of [4] (2 minutes in total, and 400^{th}, 600^{th}, 2500^{th}, 3000^{th} frames are shown). Bottom: Our synopsis (1 minute in total, and 100^{th}, 600^{th}, 1200^{th}, 1500^{th} frames are shown), by setting K to 3 in the iterative object shifting.

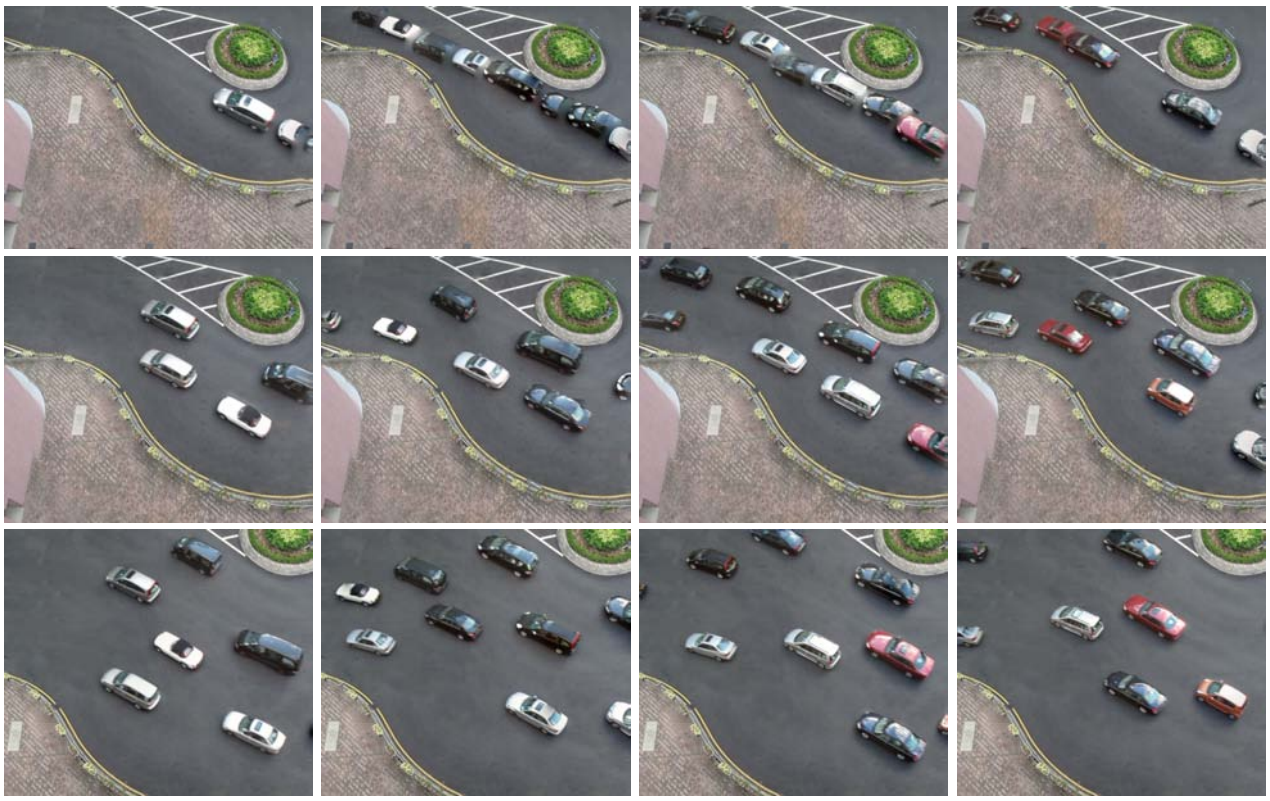


Fig. 12. Input video (3 minutes). Top: Video synopsis result of [4] (23 seconds in total, and 50^{th}, 250^{th}, 350^{th}, 450^{th} frames are shown). Middle: Our synopsis (15 seconds in total, and 50^{th}, 150^{th}, 250^{th}, 350^{th} frames are shown), by setting K to 2 in the iterative object shifting step. Bottom: Our even shorter synopsis (10 seconds in total, 10^{th}, 100^{th}, 170^{th}, 200^{th} frames are shown), by setting K to 4.

Fig. 13. Input video (30 minutes). Top: Video synopsis result of [4] (1 minute in total, and 212^{th}, 570^{th}, 855^{th}, 1477^{th} frames are shown). Bottom: Our synopsis (1 minute in total, and 212^{th}, 570^{th}, 855^{th}, 1477^{th} frames are shown).

Fig. 14. Input video (5 minutes). Top: Video synopsis result of [4] (24 seconds in total, and 150^{th}, 215^{th}, 300^{th} frames are shown). Bottom: Our synopsis (24 seconds in total, and 150^{th}, 215^{th}, 300^{th} frames are shown).

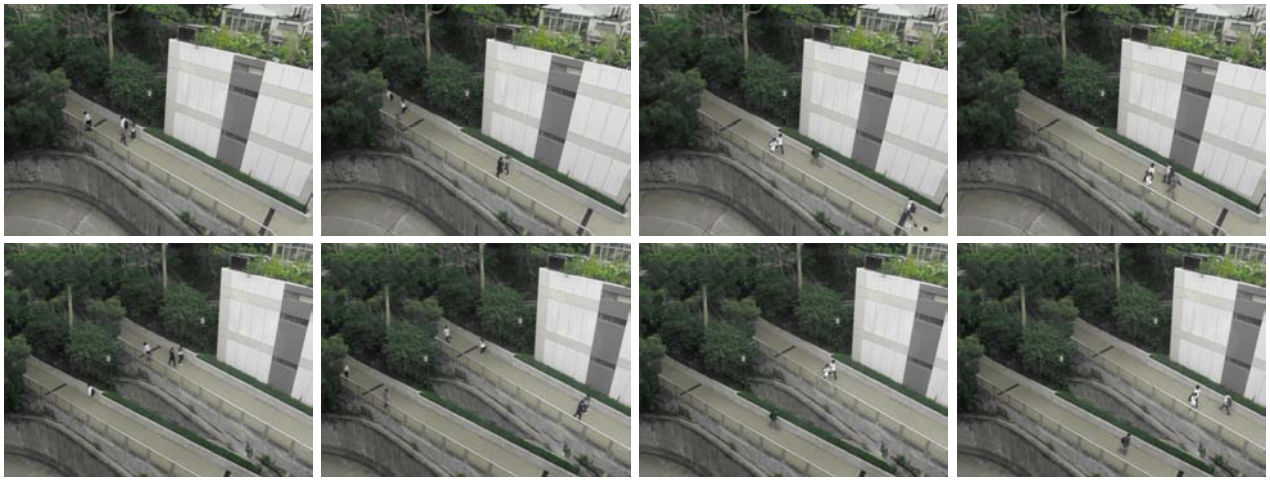


Fig. 15. The input video (10 minutes). Top: Video synopsis result of [4] (66 seconds in total, and 200^{th}, 600^{th}, 1200^{th}, 1800^{th} frames are shown). Bottom: Our synopsis (40 seconds in total, and 250^{th}, 500^{th}, 850^{th}, 900^{th} frames).

aware fashion. In this example, K is set to 3 in the iterative object shifting method.

Fig. 12 shows a traffic-circle video example (3 minutes) in which cars run in and out. In this case, all the cars are moving in the same direction. The top row shows the synopsis result of [4] which shifts the cars temporally only. If compressed too much in the time domain (23.3 seconds), the cars move too close to each other and collision artifacts occur among them. In our approach, the global spatiotemporal optimization shifts cars in both temporal and spatial free space, and we accordingly expand the movement space into the compact background. Our results in the second row show that collisions are avoided and our synopsis is shorter (15 seconds). For the bottom row, the parameters are set as: $\alpha = 1$, $\beta = 2.5$, $\gamma = 0.1$, $\epsilon = 0.5$ and $K = 4$. α and β are decreased to increase the relative weight of term E_a, which ensures that the objects remain within the synopsis. γ is decreased to move objects a little farther away, which allows us to expand more into spatial free space and to obtain an even shorter synopsis (10 seconds).

In Fig. 13, the input video is approximately 30 minutes. Change in shadows is apparent in the video. Our approach extracts a background image every minute. Based on these backgrounds, we synthesize time-varying compact backgrounds, which are then composited with the shifted objects in the spatiotemporal space. Thus, the shadow-variance phenomenon is preserved quite well in our final compact synopsis. Here, we show our video synopsis result, which is the same length as that of [4]. However, ours contains no collisions.

In Fig. 14, the input video (5 minutes) shows a narrow winding road in the woods, where people come and go very often. The top row shows the video synopsis result (24 seconds) of [4] using temporal shifts only, in which collisions among people walking in opposite directions are apparent. Using our approach, the moving objects are shifted in the global spatiotemporal video volume, which effectively avoids motion collision artifacts. In the compact background, we expand the moving space to three roads by setting K to 3. The bottom row shows our synopsis result (also 24 seconds for comparison) without motion collisions.

In Fig 15, the input video (10 minutes) shows a narrow lane often seen in hilly areas. Due to the side camera view, the moving people may cover parts of railings of the road. Since the extracted mask of moving people is not precise and is larger than the people themselves, it may contain extra railings. If we provide a wider lane in the compact background, people with extra railings would appear in the middle of lane, which causes visual artifacts. To avoid this kind of artifact, we relax the coefficient γ of cost term E_d to 0.1, which moves the objects a little farther from their original trajectories. Accordingly, two lanes are synthesized in the compact background, which is very compatible with the shifted objects. The top row shows the video synopsis (66 seconds) of [4] with serious motion collisions. The bottom row shows our shorter synopsis result (40 seconds) without any collisions.

User Study We performed a user study with 30 participants to validate the effectiveness of our spatiotemporal video synopsis method. Each time, a participant was shown two synopsis videos side by side. On the left was the result of [4] and on the right was our synopsis result. Each participant browsed the five synopsis examples shown in Fig. 11 to Fig. 15, and was asked to answer "Yes" or "No" to the following five questions for each example: (1) Do you think the synopsis on the right is interesting? (2) Do you like the collisions in the synopsis on the left? (3) Do you care more about the objects themselves than the relations between objects and the environment? (4) Can you recognize the objects in the synopsis on the left? (5) Can you recognize the objects in the synopsis on the right? Then they were shown the source video and were asked to answer this question: (6) Is the synopsis on the right better than the synopsis on the left for presenting the source video?

For each question, let $A_{ij} = 1$ if the i^{th} user answered "yes" to the j^{th} example, otherwise $A_{ij} = 0$. We use the following equation to compute the rate of "Yes" responses to this question, which reflects the users' opinions:

$$R = (\sum_{i=1}^{30}\sum_{j=1}^{5} A_{ij})/150 * 100\%. \quad (18)$$

The "Yes" rates for the six questions were 82.7%, 4%, 66.7%, 38.7%, 96% and 84%, respectively. From this testing data, we can see that most users think our work is more interesting and our results are better than that of [4].

By directly operating on the extracted objects, Pritch et al. [4] find and remove redundancies in a much finer grained style than frame-based or synthesis-based synopsis methods. The algorithm used in [4] is very useful for compressing surveillance videos and essentially inspires our work. Since it compresses objects only in the time scale, one of its advantages is that it does not need to adjust the background of the scene. Thus, it can process videos of complex scenes such as airports, flyovers, and crossroads. However, for objects moving on the same path, this method compacts them in a spatially narrow and temporally short space, leading to numerous overlaps among objects and making it difficult for users to understand what happened. Our method can handle surveillance videos with narrow movement space and with greater spatial free space, which can't be handled well by [4]. Our work is complementary to [4] and enriches the object-based synopsis methods.

Limitation When utilizing both the temporal and spatial spaces to eliminate collisions, our method may fail to produce a compact video synopsis for scenes with crowded activity in both the spatial and temporal domains. For example, our method can't handle videos full of paths with objects moving on them, since there is no spatial free space to expand into. In such situations,

methods of selecting important objects and discarding undesired ones can be considered. The use of different selection methods can result in different synopses. We will further investigate these measurements and extend our synopsis framework to effectively process crowd-scene videos. The second limitation is that the spatiotemporal optimization stage has no interaction with the compact background synthesis stage. As a result, even if the spatiotemporal optimization stage works well, the background synthesis may fail to produce a consistent result. Fig. 15 shows one example reflecting this limitation. To solve this problem, the user needs to go back to the first stage and adjust parameters to obtain better object shifting. The patch relocation in our compact background synthesis stage may fail if the original background has strong structure. Though our proposed multilevel and local patch relocation methods alleviate this problem to a certain extent, we still need a more structure-aware image editing framework in the future.

6 Conclusion and Future Work

Video synopsis is a useful tool for summarizing long surveillance videos captured by digital cameras. In this paper, we propose a novel global video synopsis approach that shifts objects in spatiotemporal space. Our method eliminates object collisions to create shorter, more visually appealing synopses. We expand the movement space of shifted objects into the compact background in an environmentally context-aware fashion, avoiding visual artifacts such as cars running off the roads or people walking on water. The experimental results show that our proposed method is a practical and efficient video synopsis tool.

In the future, we will investigate how to use our synopsis method as a video editing tool. We will also investigate more structure-aware methods for synthesizing compact backgrounds from videos. Currently, our proposed method works on videos with static backgrounds. We will extend it to handle videos captured by moving cameras. This is a more challenging topic, due to the dynamic backgrounds, and the different types of camera motion, such as panning, zooming, and jittering, etc.

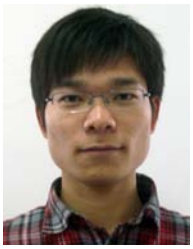
Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and insightful suggestions, and thank to Isaac Liao for proofreading the manuscript. This work was partly supported by the National Basic Research Program of China (No. 2012CB725303), NSFC (No. 61070081), RGC research grants (ref. 416007, 416311), UGC direct grant for research (no. 2050454, 2050485), the Open Project Program of the State Key Lab of CAD&CG (Grant No. A1208), Luojia Outstanding Young Scholar Program of Wuhan University, the Project of Science and Technology Plan for Zhejiang Province (Grant No. 2012C21004), and the Fundamental Research Funds for the Central Universities. Chunxia Xiao is the corresponding author.

References

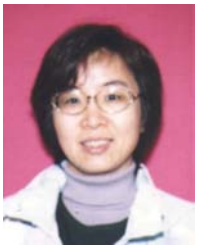
[1] Y.Li, T. Zhang, and D. Tretter, "An overview of video abstraction techniques," HP Laboratories Palo Alto, 2001.
[2] H. Kang, X. Chen, Y. Matsushita, and X. Tang, "Space-time video montage," in Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, vol. 2. IEEE, 2006, pp. 1331-1338.
[3] B. Truong and S. Venkatesh, "Video abstraction: A systematic review and classification," ACM Transactions on Multimedia Computing, Communications, and Applications(TOMCCAP), vol. 3, no. 1, p. 3, 2007.
[4] Y. Pritch, A. Rav-Acha, and S. Peleg, "Nonchronological video synopsis and indexing," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1971-1987, 2008.
[5] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani, "Summarizing visual data using bidirectional similarity," in Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. IEEE, 2008, pp. 1-8.
[6] C. Kim and J. Hwang, "An integrated scheme for object-based video abstraction," in Proceedings of the eighth ACM international conference on Multimedia. ACM, 2000, pp. 303-311.
[7] J. Nam and A. Tewfik, "Video abstract of video," in Multimedia Signal Processing, 1999 IEEE 3rd Workshop on. IEEE, 1999, pp. 117-122.
[8] E. Bennett and L. McMillan, "Computational time-lapse video," in ACM Transactions on Graphics (TOG), vol. 26, no. 3. ACM, 2007, p. 102.
[9] A. Rav-Acha, Y. Pritch, and S. Peleg, "Making a long video short: Dynamic video synopsis," in Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, vol. 1. IEEE, 2006, pp. 435-441.
[10] Y. Pritch, A. Rav-Acha, A. Gutman, and S. Peleg, "Webcam synopsis: Peeking around the world," in Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on. IEEE, 2007, pp. 1-8.
[11] T. Liu, X. Zhang, J. Feng, and K. Lo, "Shot reconstruction degree: a novel criterion for key frame selection," Pattern recognition letters, vol. 25, no. 12, pp. 1451-1457, 2004.
[12] X. Zhu, X. Wu, J. Fan, A. Elmagarmid, and W. Aref, "Exploring video content structure for hierarchical summarization," Multimedia Systems, vol. 10, no. 2, pp. 98-115, 2004.
[13] N. Petrovic, N. Jojic, and T. Huang, "Adaptive video fast forward," Multimedia Tools and Applications, vol. 26, no. 3, pp. 327-344, 2005.
[14] C. Taskiran, Z. Pizlo, A. Amir, D. Ponceleon, and E. Delp, "Automated video program summarization using speech transcripts," Multimedia, IEEE Transactions on, vol. 8, no. 4, pp. 775-791, 2006.
[15] Y. Li, S. Narayanan, and C. Kuo, Movie content analysis, indexing and skimming via multimodal information. Kluwer Academic Hingham, MA, 2003.
[16] J. Wu, Perspectives on content-based multimedia systems. Springer Netherlands, 2000.
[17] J. Ouyang, J. Li, and Y. Zhang, "Replay boundary detection in mpeg compressed video," in Machine Learning and Cybernetics, 2003 International Conference on, vol. 5. IEEE, 2003, pp. 2800-2804.
[18] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman, "Patchmatch: A randomized correspondence algorithm for structural image editing," in ACM Transactions on Graphics (TOG), vol. 28, no. 3. ACM, 2009, p. 24.
[19] C. Xiao, M. Liu, Y. Nie, and Z. Dong, "Fast exact nearest patch match for patch-based image editing and processing," Visualization and Computer Graphics, IEEE Transactions on, no. 99, pp. 1-1, 2011.
[20] V. Kwatra, I. Essa, A. Bobick, and N. Kwatra, "Texture optimization for example-based synthesis," ACM Transactions on Graphics (TOG), vol. 24, no. 3, pp. 795-802, 2005.
[21] L. Wei and M. Levoy, "Fast texture synthesis using tree-structured vector quantization," in Proceedings of the 27th annual conference on Computer graphics and interactives techniques. ACM Press/Addison-Wesley Publishing Co., 2000, pp. 479-488.
[22] C. Xiao, Y. Nie, W, Hua, and W. Zheng, "Fast multi-scale joint bilateral texture upsampling," The Visual Computer, vol. 26, no. 4, pp. 263-275, 2010.

[23] Y. Wexler, E. Shechtman, and M. Irani, "Space-time video completion," in Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, vol. 1. IEEE, 2004, pp. I-120.
[24] J. Sun, L. Yuan, J. Jia, and H. Shum, "Image completion with structure propagation," ACM Transactions on Graphics (ToG), vol. 24, no. 3, pp. 861-868, 2005.
[25] C. Xiao, S. Liu, H. Fu, C. Lin, C. Song, Z. Huang, F. He, and Q. Peng, "Video completion and synthesis," Computer Animation and Virtual Worlds, vol. 19, no. 3-4, pp. 341-353, 2008.
[26] T. Cho, M. Butman, S. Avidan, and W. Freeman, "The patch transform and its applications to image editing," 2008.
[27] T. Cho, S. Avidan, and W. Freeman, "The patch transform," IEEE transactions on pattern analysis and machine intelligence, pp. 1489-1501, 2010.
[28] A. Jain, R. Duin, and J. Mao, "Statistical pattern recognition: A review," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 22, no. 1, pp. 4-37, 2000.
[29] S. Cohen, "Background estimation as a labeling problem," in Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, vol. 2. IEEE, 2005, pp. 1034-1041.
[30] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," in ACM Transactions on Graphics (TOG), vol. 22, no. 3. ACM, 2003, pp. 313-318.
[31] J. Sun, W. Zhang, X. Tang, and H. Shum, "Background cut," Computer Vision-ECCV 2006, pp. 628-641, 2006.
[32] P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in Proc. 2nd European Workshop on Advanced Video Based Surveillance Systems, vol. 25, 2001.
[33] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" IEEE transactions on pattern analysis and machine intelligence, pp. 147-159, 2004.
[34] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 23, no. 11, pp 1222-1239, 2001.
[35] J. Yedidia, W. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," Exploring artificial intelligence in the new millennium, vol. 8, pp. 236-239, 2003.
[36] S. Avidan and A. Shamir, "Seam carving for content-aware image resizing," in ACM Transactions on Graphics (TOG), vol 26, no. 3. ACM, 2007, p. 10.



Yongwei Nie received the BS degree from the School of Computer, Wuhan University, in 2009. Currently, he is working toward the PHD degree at the School of Computer, Wuhan University, China. His research interests include image and video editing, and computational photography. Contact him at nieyongwei@gmail.com.

Chunxia Xiao received the BSc and MSc degrees from the Mathematics Department of Hunan Normal University in 1999 and 2002, respectively, and the PhD degree from the State Key Lab of CAD & CG of Zhejiang University in 2006. Currently, he is a professor at the School of Computer, Wuhan University, China. From October 2006 to April 2007, he worked as a postdoc at the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. His main research interests include image and video processing, digital geometry processing, and computational photography. Contact him at cxxiao@whu.edu.cn.



Hanqiu Sun is an associate professor at the Chinese University of Hong Kong. Her research interests include virtual & augmented reality, interactive graphics/animation, hypermedia, mobile image/video processing and navigation, touch-enabled simulations. Sun has an MS in electrical engineering from University of British Columbia and PhD in computer science from University of Alberta, Canada. Contact her at hanqiu@cse.cuhk.edu.hk.

Ping Li is a Ph.D. candidate in Department of Computer Science & Engineering, the Chinese University of Hong Kong. His research interests are image/video processing and creative media, including image/video retexturing, stylization, colorization, video summarization, and GPU acceleration. He received his B.Eng. in Computer Science & Technology from China Jinan University. Contact him at pli@cse.cuhk.edu.hk.