Object Movements Synopsis via Part Assembling and Stitching

Yongwei Nie, Hanqiu Sun, Ping Li, Chunxia Xiao, and Kwan-Liu Ma, *Fellow, IEEE*

Abstract—Video synopsis aims at removing video's less important information, while preserving its key content for fast browsing, retrieving, or efficient storing. Previous video synopsis methods, including frame-based and object-based approaches that remove valueless whole frames or combine objects from time shots, cannot handle videos with redundancies existing in the movements of video object. In this paper, we present a novel part-based object movements synopsis method, which can effectively compress the redundant information of a moving video object and represent the synopsized object seamlessly. Our method works by part-based assembling and stitching. The object movement sequence is first divided into several part movement sequences. Then, we optimally assemble moving parts from different part sequences together to produce an initial synopsis result. The optimal assembling is formulated as a part movement assignment problem on a Markov Random Field (MRF), which guarantees the most important moving parts are selected while preserving both the spatial compatibility between assembled parts and the chronological order of parts. Finally, we present a non-linear spatiotemporal optimization formulation to stitch the assembled parts seamlessly, and achieve the final compact video object synopsis. The experiments on a variety of input video objects have demonstrated the effectiveness of the presented synopsis method.

Index Terms—Video synopsis, part assembling, part stitching, belief propagation, MRF optimization

1 Introduction

The fast advances in digital video acquisition lately have made numerous raw video data. Video synopsis, which generates short summary of input videos, has become an active research topic in graphics and vision communities. The essential idea in video synopsis is to preserve important contents in a short video representation while eliminating most of the redundant or less important information. A variety of video synopsis methods have been introduced, which can be roughly divided into two categories: the frame-based approaches (see [1] as a survey) and the object-based approaches [2], [3], [4], [5]. In the former approaches, key frames are extracted and viewed as the basic building blocks of synopsis, while other less important frames are removed as redundancies. Alternatively, the latter methods treat each of the moving objects as a 3D tube, and eliminate spatiotemporal redundancies between objects by shifting the tubes along time axis.

Due to the complexity of dynamic videos, we observe that both the frame-based and object-based synopsis approaches cannot effectively handle videos with redundancies in the moving objects themselves. In this paper, we will focus on the movements of a single video object, and try to eliminate the redundancies existing in the object movements, which helps to produce more compact video synopsis. Our basic idea is to work at the level of object part, and to remove the non-moving parts which are considered as redundancies in this paper.

At the level of object part, we observe two kinds of part-based redundancies during the movement of an object: inter-part redundancies and inner-part redundancies. In an input video, an object often moves a part first, and then moves another part. Assuming the moving parts are important content that should be preserved, the corresponding non-moving parts at the same time can be considered as inter-part redundancies, and can be removed when they bring much less information. Besides the inter-part redundancies, we also find inner-part redundancies existing in a moving part itself, such as repeated or slow movements of the part. By eliminating both the inter-part and inner-part redundancies at the part level in an optimal procedure, we can produce a more compact synopsis for object movements.

In this paper, we use "object movement" to indicate a frame of a moving object in an input video, and use "part movement" to indicate a frame of a part of a moving object. To remove part redundancies, the first difficulty is how to partition each object movement into several semantic part movements. This is not a problem in many computer generated animations or cartoons, since in those videos the object parts are known and can be segmented perfectly. For natural videos, it may cost the user much time on segmenting object parts. However, that is worthwhile since the user only needs to segment object once, but can watch the synopsis many times.

- *Y. Nie and C. Xiao are with the Computer School of Wuhan University, Wuhan, HuBei 430072, China. E-mail: nieyongwei@gmail.com, cxxiao@whu.edu.cn.*
- *H. Sun is 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.*
- *P. Li is with the Department of Mathematics and Information Technology, the Hong Kong Institute of Education, Tai Po, NT, Hong Kong. E-mail: pli@ied.edu.hk.*
- *K.-L. Ma is with the Department of Computer Science, University of California-Davis, CA 95616-8562. E-mail: ma@cs.ucdavis.edu.*

Manuscript received 8 Jan. 2013; revised 19 Nov. 2013; accepted 16 Dec. 2013. Date of publication 8 Jan. 2014; date of current version 30 July 2014. Recommended for acceptance by D. Schmalstieg. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TVCG.2013.2297931

Based on the above observations, we present a novel object movements synopsis method to effectively remove the redundancies during the movements of a video object. Our method works in the following three main steps: object movement partition, part movement assembling, and part movement stitching. We first present a method to extract part movement sequences by partitioning each object movement into several semantic parts. We then optimally select the same number of the most important movements from each part movement sequence and assemble them together frame by frame to obtain more compact object movements. We cast this as a part movement assembling problem on MRF and solve it by Loopy Belief Propagation (LBP). Since only important movements of each part are selected, both the inter-part and inner-part redundancies can be eliminated successfully. As the assembled parts of a synopsized object movement usually come from different frames, gaps may occur between adjacent parts. To eliminate the gap artifacts, we shift all of the parts in their spatial space using a part movement stitching optimization, which seamlessly stitches the spatially adjacent parts.

By assembling object parts from different frames, we synthesize new object movements that do not exist in the input video into the synopsis. We demonstrate in the experiments that each introduced movement is a good representation of many original object movements in the input video. The method presented in this paper is a step toward promoting more finer-grained redundancy classification and removal at object part level. Our method facilitates online video browsing, indexing, retrieval, and big data archiving.

In the following, we briefly outline the related work in Section 2. We give a general overview of the proposed approach in Section 3. Then we give the technical details of the presented method in Section 4, where Section 4.1 describes object partition, Section 4.2 introduces the proposed part movement assembling optimization, and Section 4.3 describes the part movement stitching optimization. The video synopsis results and discussions are given in Section 5. The conclusion and future work goes to Section 6.

2 Related Work

Frame-based video abstraction has received extensive research during last two decades. This kind of methods consider frame as the basic building block that cannot be discomposed. There are two basic forms of video abstracts: keyframes and video skims. The keyframe-based abstraction methods [6], [7] extracted a collection of salient images from the input video. Researchers improved the descriptors and criteria used in keyframe selection process, such as taking high-level content analysis into account [8], [9] to further ensure the accuracy of the process. Other than selecting only one frame once, video skimming methods identify and extract important video clips [10], [11]. The clips are then connected by cut or gradual effect. The major advantage of frame-based methods is that they are usually simple and efficient, which makes them be widely used in image/video processing industry. As the frame-based video abstraction methods abandon a frame as a whole, they usually suffer from losing fast activities.

To utilize empty space of frames, many methods directly work on the moving objects. [2], [3] extracted objects out of video and treated them as 3D tubes which were then shifted along time axis to remove temporal redundancies among objects. Nie et al. [4] shifted objects not only in the temporal domain but also in the spatial domain, this method expanded more motion space for objects and achieved more compact video synopsis. Lu et al. [5] edited timelines of objects for accelerating or slowing objects while preserving the original object interactions. Schödl and Essa [12] captured movements of an object, and rearranged the original chaotic movements to obtain a sequence of regular and meaningful movements of the object. By precisely identifying interesting objects in videos, the object-based methods can eliminate spatiotemporal redundancies, to preserve interesting moving objects rather than static scenes. However, object-based methods cannot effectively handle videos with redundancies existing during the movement of a single object. Our method works at part level, which is more finer-grained than the object-based methods and gives us more flexibility to remove finer-grained redundancies.

Space-time video montage [13] and image/video collage [14], [15] can also be seen as one kind of object-based video synopsis methods to a certain extent. In these methods, the building blocks are video portions. In the short summary, a minimum cost boundary between two adjacent portions is computed to stitch them together. Seams may be apparent when texture information is less along those boundaries.

Synthesis-based video summarization [16], [17], [18] introduced a bidirectional similarity for measuring whether the summary is visually similar to the input source video. The good summary should contain as much as possible important contents from the input video, but not introduce contents that do not show in the source video. As the best matches of cubes in the input video and summary must be found and stored, the processing is quite time and memory consuming. Based on [16], Barnes et al. [19] presented a multi-scale tapestries summarization approach by synthesizing image that is continuous in spatial and scale dimensions.

Salient stills is a very interesting technique that aggregates salient objects extracted out of a moving image sequence together into only one image frame [20]. It shows that for a short video clips, much of the captured information is repeated in several frames. They show the same background information, and spread foreground objects frames together on one picture. Dynamic stills introduced in [21] extends the existing work by developing functions to rotate, translate and scale both the background and the foreground information, which effectively presents self-occluding activities. A generalization of salient still pictures is stroboscopic effects [2] that display multiple dynamic appearances of a single object. Recently, Correa and Ma [22] used this technology in their dynamic video narratives framework for depicting the motions of actors over time simultaneously. Video panoramic or dynamosaics can also reduce the length of input videos [23], [24] based on the key observation that the adjacent frames shot by panning camera have much common information, thus can be stitched together seamlessly by precise alignment. Our method shares some similar ideas with the above methods that each

frame of the synopsis conveys information that may come from more than one input frame.

To the best of our knowledge, there is no previous work condensing object movements. Assa et al. [25] developed an action synopsis method. But this method only selected key motions of objects from a motion sequence and illustrated them in a still image. Instead, we synthesize more informative object movements by recombining object parts. Structure recovery at part level has appeared in computer graphics research. For example, Shen et al. [26] presented a method for recovering Kinect acquired 3D structures by part assembly. However, they built an extra 3D part data set, while our method selects parts from the source video itself.

Many methods have been proposed for segmenting and tracking foreground objects from videos. The hard segmentation, such as graph-based methods [27], [28], can obtain good results with non-trivial user interactions. Zhang et al. [29] presented a method to extract and complete object layers from cartoon animations. In image/video editing tasks [30], [31], [32], [33], interactive brush is used to select object in video. Soft segmentation like video matting [34] provides a better alternative if one needs to paste the extracted object into a new background, but it usually needs appropriate trimaps as constraints. Agarwala et al. [35] presented a novel approach of rotoscoping which effectively tracks contours in video sequences with some user interaction.

3 Overview

We give an overview of the proposed object movements synopsis method in Fig. 1. Three stages are taken to produce the final synopsis result: object partition, part movement assembling, and part movement stitching. In the first stage (Section 4.1), we extract an object movement sequence from an input video (Fig. 1a), and partition each object into several semantic parts, which produces several part movement sequences (Fig. 1b). The partition boundaries between adjacent parts are kept for using in the next two stages.

In the second stage (Section 5), we select the same number of part movements from each part sequence. These moving parts are then assembled frame by frame to form the synopsized object movements. We select the most important part movements while considering the other two requirements: (1) the assembled part movements should be spatially compatible with each other, which is implemented as the similarity between partition boundaries of the parts; (2) the chronological order of parts should be preserved, i.e, a part in behind cannot appear in front. We cast this as a part movement assignment problem on MRF and solve it by Loopy Belief Propagation. In Fig. 1b, the blue ones are selected from the "Right Arm", "Torso" and "Left Arm" movement sequences and are assembled together to form a synopsized object movement (Fig. 1c). Since only the most important moving parts are selected, part-level redundancies, i.e., the non-moving parts, are eliminated successfully.

However, in the results produced by the part movement assembling optimization, assembled parts may not exactly stitch to each other (Fig. 1c), since the parts are collected from different frames of the input video. In the third stage (Section 6), we stitch the assembled parts to eliminate the gaps among them using a part movement stitching optimization, which guarantees producing spatiotemporally continuous and consistent synopsis (Fig. 1d).

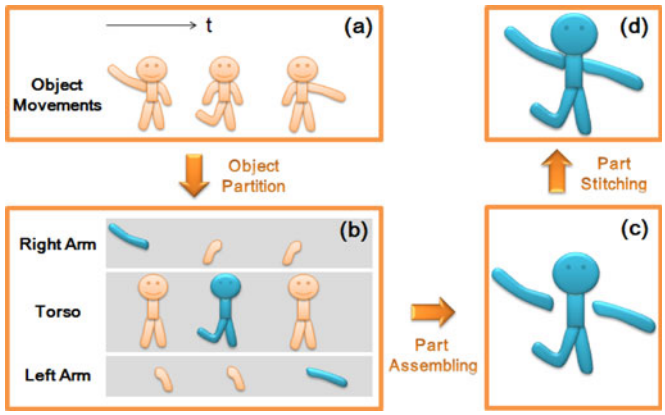


Fig. 1. Overview of the object movements synopsis system. (a) The input object movement sequence. (b) Three part movement sequences: the right arm, the torso and the left arm. (c) The part movement assembling optimization assembles the most important movements of different parts together to produce a synopsized object. (d) The assembled parts are stitched together by using a spatiotemporal part movement stitching optimization.

4 Object Movements Synopsis

In this section, we describe in detail the three stages used in our object movements synopsis system, i.e, object partition, part movement assembling optimization, and part movement stitching optimization.

4.1 Object Partition

First, we partition each object movement into several semantic parts. We compute and keep the partition boundaries between adjacent parts, which will be used in the next two stages. We also compute an importance value for each moving part. The importance value of a part movement determines the opportunity of the movement to be preserved in the synopsis.

Object Partition. It is well known that automatically segmenting and tracking video objects is still a hard problem. Hard segmentation methods, such as [27], [28], require much user interaction to obtain desirable results. In this paper, we employ a method combining rotoscoping [35] and matting [36] to partition object and track object parts, which requires limited user inputs. Rotoscoping [35] is a technique which tracks object contours in video. The user draws and controls curves around an object part on keyframes, and the curves on the in-between frames will be automatically interpolated (see Fig. 2). With the help of the rotoscoping curves, we generate trimaps of an object part. We then extract the part by matting method of [36]. Our approach saves user interaction in two ways: (1) the user only needs to draw control curves on keyframes but not all the frames of the video, and (2) the control curves do not need to match the object contour exactly, thus avoiding pixel-wise adjusting of the control curves.

Partition Boundaries. After partitioning an object into several parts, we compute and keep partition boundaries between adjacent parts. We propose a simple and efficient

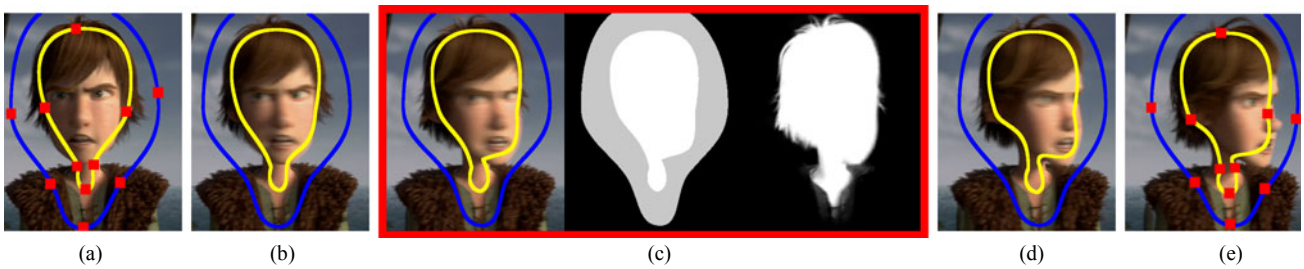


Fig. 2. (a) and (e) are two keyframes where the user draws controlling curves around the head contours. The curves on the in-between frames (b)-(d) are automatically interpolated by rotoscoping [35]. For the third frame (c), we give the trimaps generated from the curves, and also give the matting result.

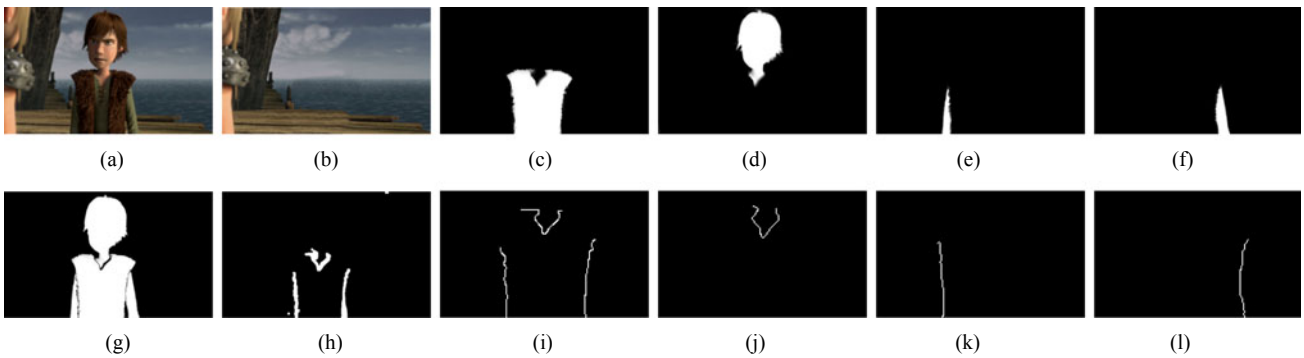


Fig. 3. Partition boundary computing. (a) Input object movement. (b)-(e) Four parts: torso, head, right arm, and left arm. (f) The filtered matte of the object. (g) The union of filtered (b), (c), (d) and (e). (h) The seams between adjacent parts. (i)-(l) Partition boundaries of the four parts.

method based on several Boolean operations to address this problem, which is illustrated in Fig. 3. Fig. 3a shows an input object movement. To obtain the background in Fig. 3b, we extract mattes of the foreground object. Then, we remove the object and repair the hole by structure completion method [37]. Four parts (torso, head, right arm, and left arm) of the object that are filtered by a threshold δ (224 out of 255) are shown from (c) to (f). The filtering operation sets those mattes less than δ to zero. We combine the filtered mattes from (c) to (f) together to obtain (g). Since small mattes along the boundaries of parts are set to zero, we see black seams in (g) that separate different parts. We extract the seams and show them in (h). An "and" operation is then used between (h) and mattes of each part movement to get the partition boundary of each part. Finally, we apply robust thinning algorithm [38] to make each partition boundary having only one one-pixel width, as illustrated in (i)-(l).

Importance Estimation. Our method preserves the most important moving parts in the synopsis. Thus we must compute an importance value for each moving part. By assuming the faster a part moves the more important the part is, we use the state-of-art optical flow method of [39] to determine a part movement's importance. Given the flow vector of each pixel, the importance value of a moving part is computed as the average flow vector length of the part. Fig. 4 shows importance variation over time of the four parts in Fig. 3. The peaks indicate the most important part movements that should be preserved in the synopsis.

4.2 Part Movement Assembling Optimization

Once we have partitioned the object movements, and obtained a sequence of movements for each part, we then select the same number of movements from each part sequence, and assembled them together frame by frame to produce a synopsis. Our selection scheme should satisfy the following properties:

1. The synopsis length M must be shorter than the length N of the input video.
2. The most important moving parts should be preserved in the synopsis.
3. Spatially assembled parts should be compatible with each other.
4. The chronological order must be preserved for the same part, while the order between different parts can be disrupted.

This selection problem is solved by a part movement assignment optimization on MRF. We define a 3D graph $G = \{V, E\}$ according to the connection relationships between object parts, where V is a set of nodes, and E is a set of edges linking the nodes. Let K be the number of parts an object is partitioned to, there will be K types of nodes each representing a kind of part. There are two types of edges, the spatial edges linking different types of nodes on the same frame of the synopsis, and the temporal edges linking the same type of nodes on adjacent frames. Fig. 5

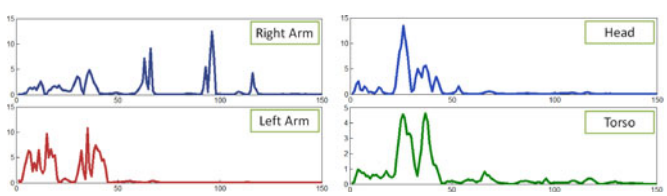


Fig. 4. Importance variation over time of the four parts in Fig. 3. The peaks indicate the most important part movements that should be preserved in the synopsis.

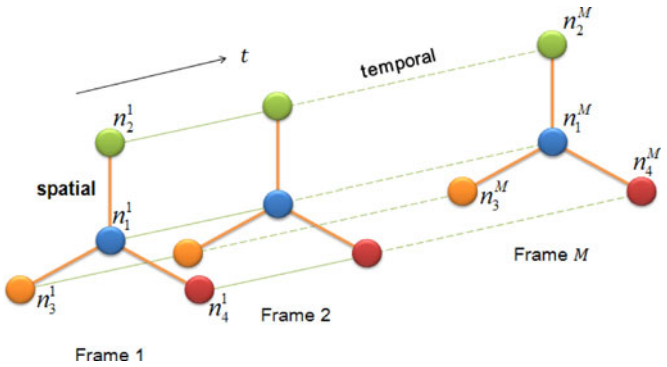


Fig. 5. The MRF graphical model of the presented method. Each node will be assigned a corresponding part movement. The orange lines link spatially adjacent nodes, and the gray lines link temporally adjacent nodes.

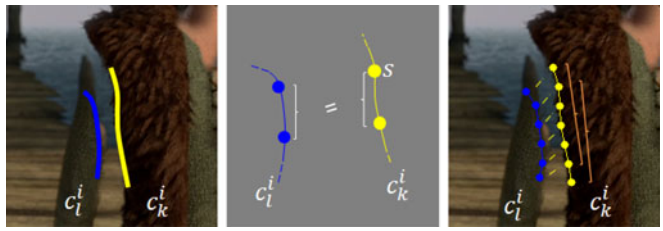


Fig. 6. (a) The partition boundaries of two assembled moving parts. (b) We sample a point every 10 pixels on each boundary. (c) Different boundary may have different length and thus different number of sampling points. Each part with the same length of the longer boundary is compared with the shorter boundary, and the most similar part is selected.

shows a MRF graph for the object in Fig. 3, where on each frame a torso part node (blue) links to the head node (green), the right arm node (orange), and the left arm node (scarlet).

In this paper, we use k and l to index different kinds of MRF nodes or the corresponding parts, and i and j to index video frames. Let n_k^i ($k \in [1, K]$ and $i \in [1, M]$) be the kth type of node on the ith frame of the synopsis, $x_k \in [1, N]$ be the index of a moving part from the kth kind of part sequence, we use x_k^i to represent the index of a moving part assigned to a node n_k^i. Our aim is to assign all the nodes the best indices X that satisfies the above four properties. We define the joint probability of this part movement assignment problem as:

$$P(X) \propto \prod_{i,k} \Phi(x_k^i) \prod_{i,k,\binom{j}{l} \in \mathcal{N}\binom{i}{k}} \Psi(x_k^i, x_l^j), \tag{1}$$

where $\binom{j}{l}$ is short for the lth type of node n_l^j on the jth frame of the synopsis, and $\binom{j}{l} \in \mathcal{N}\binom{i}{k}$ means any node n_l^j that is spatially or temporally adjacent to node n_k^i.

$\Phi(x_k^i)$ is the importance term that keeps important moving parts into the synopsis. Let $\chi(\cdot)$ be the importance value of a moving part estimated in Section 4.1, $\Phi(x_k^i)$ is computed as the convolution of $\chi(\cdot)$ and a Gaussian filter e:

$$\Phi(x_k^i) = \chi(x_k^i) \cdot e^{-\frac{1}{2}\left(\frac{x_k^i - \frac{M}{N}i}{\sigma_\Phi}\right)^2}. \tag{2}$$

We filter the importance function $\chi(\cdot)$ by e to ensure that the locally most important moving part are selected for each node rather than the globally most important one. σ_Φ is set as 20 in all the experiments of this paper.

Term $\Psi(x_k^i, x_l^j)$ preserves consistency between moving parts x_k^i and x_l^j that are placed at spatially or temporally adjacent nodes n_k^i and n_l^j, which is defined as:

$$\Psi(x_k^i, x_l^j) = \begin{cases} S(x_k^i, x_l^i), if\ i = j & l \in \mathcal{SN}(k), \\ T(x_k^i, x_k^j), if\ k = l & j \in \mathcal{TN}(i), \end{cases} \tag{3}$$

where, in the spatial domain, $\Psi(x_k^i, x_l^j)$ is defined as $S(x_k^i, x_l^i)$ which encodes compatibility between assembled parts; while in the temporal domain, $\Psi(x_k^i, x_l^j)$ is defined as $T(x_k^i, x_k^j)$ that penalizes the reversing of the chronological order of the parts assigned to the kth type of node. $\mathcal{SN}(k)$ is the spatially adjacent nodes of the kth type of node, and $\mathcal{TN}(i) = \{i-1, i+1\}$ represents temporally adjacent frames of the ith frame.

Let c_k^i and c_l^i be the partition boundaries of moving parts x_k^i and x_l^i, the spatial compatibility term $S(x_k^i, x_l^i)$ is implemented as the shape similarity between the two partition boundaries, which is inversely proportional to the boundaries' length difference $D_L(c_k^i, c_l^i)$ and curvature difference $D_C(c_k^i, c_l^i)$. Fig. 6a shows such two partition boundaries. We sample a point s every 10 pixels on each boundary (Fig. 6b), and compute the length of a boundary as the sum of the distances between any two adjacent points. When computing curvature difference $D_C(c_k^i, c_l^i)$, since different boundaries may have different length and thus different number of sampling points, we compare each part with the same length of the longer boundary with the shorter boundary (Fig. 6c), and select the most similar part. The curvature difference $D_C(c_k^i, c_l^i)$ of the two boundaries is computed as a sum of point curvature differences between the shorter boundary and the selected part. The spatial compatibility term is then defined as:

$$S(x_k^i, x_l^i) = e^{-\frac{1}{2}\left(\frac{D_L(c_k^i, c_l^i)}{\sigma_L}\right)^2} \cdot e^{-\frac{1}{2}\left(\frac{D_C(c_k^i, c_l^i)}{\sigma_C}\right)^2}, \tag{4}$$

where σ_L is 25 in this paper, and σ_C is set as 3.

The chronological term $T(x_k^i, x_k^j)$, where $j \in \{i-1, i+1\}$, is used to keep the chronological order of the moving parts assigned to the kth type of nodes, which is achieved by constraining $x_k^i < x_k^{i+1}$ frame by frame. We also require that x_k^i should not differ too much from x_k^{i+1}. Thus, $T(x_k^i, x_k^{i+1})$ is modeled as:

$$T(x_k^i, x_k^{i+1}) = \begin{cases} e^{-\frac{1}{2}\left(\frac{x_k^{i+1} - x_k^i}{\sigma_T}\right)^2}, & if\ x_k^i < x_k^{i+1}, \\ 0, & else. \end{cases} \tag{5}$$

$T(x_k^i, x_k^{i-1})$ can be computed as $T(x_k^{i-1}, x_k^i)$. A large σ_T allows x_k^i and x_k^{i+1} to differ more. We set $\sigma_T = 30$ in this paper.

We obtain a synopsis that best satisfies the above four properties by maximizing (1), which is approximately solved by the Loopy Belief Propagation [40]. The LBP algorithm maximizes (1) by approximately computing the marginal probability of the assignment of moving part x_k^i to node n_k^i. The basic mechanism of LBP is iteratively receiving messages from its neighboring nodes and then sending updated messages back to the neighborhoods. Let n_k^i and n_l^j

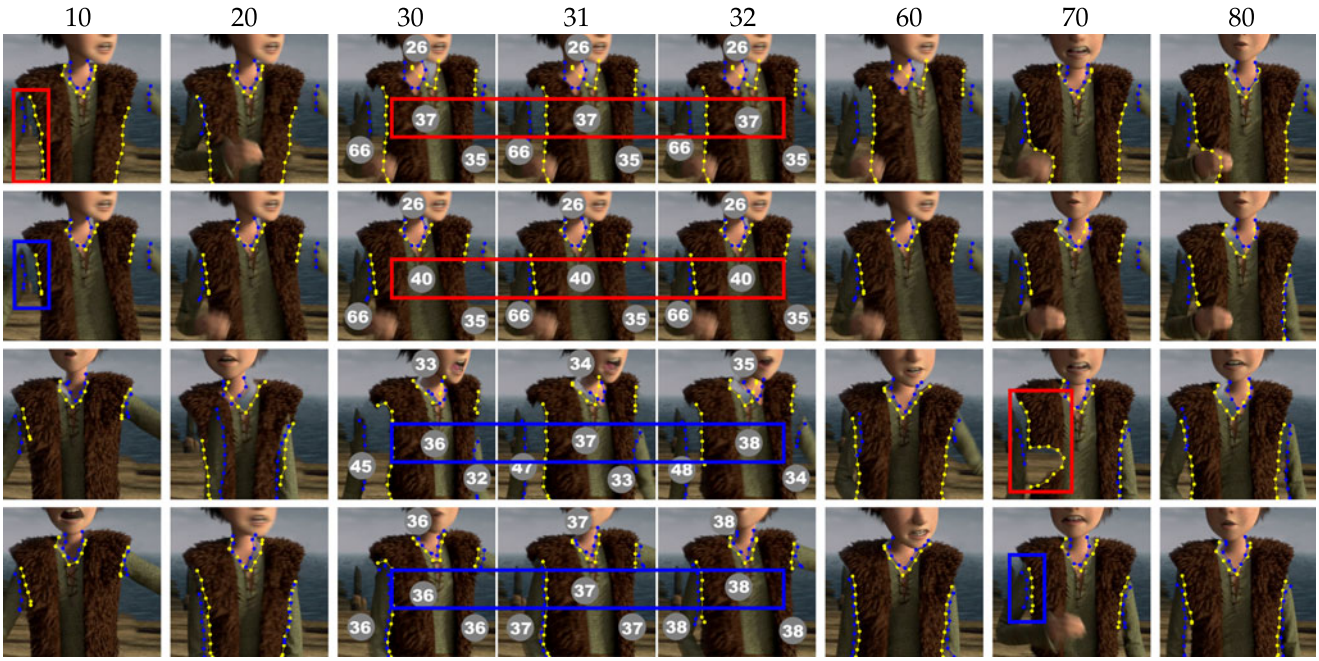


Fig. 7. Ablative analysis of the potentials in Equation (1). The input video with 151 frames is condensed to a synopsis with 100 frames. Eight frames of the synopsis are shown. *First row:* only importance term is used. The most important moving parts are kept. However, without other potentials, there are two problems: (a) spatially assembled moving parts are not compatible with each other (first red box); and (b) a locally most important moving part is assigned to several temporally consecutive nodes (second red box). *Second row:* importance term and the spatial compatibility term are used. Problem (a) is solved, but (b) still exists. *Third row:* importance term and the chronological term are used. In contrast with the second row, problem (b) is solved, but (a) exists. *Fourth row:* all the three terms are used. We achieve a good synopsis result without both problem (a) and (b).

be two spatially or temporally adjacent nodes, the message $m_{n_l^j,n_k^i}$ from n_l^j to n_k^i is:

$$m_{n_l^j,n_k^i}(x_k^i) = \sum_{x_l^j} \Psi(x_k^i, x_l^j)\Phi(x_l^j) \prod_{n\in\mathcal{N}(n_l^j)\setminus n_k^i} m_{n,n_l^j}(x_l^j). \quad (6)$$

The $\mathcal{N}(n_l^j)\setminus n_k^i$ is the spatiotemporal neighborhoods of node n_l^j except n_k^i. With iterative message updating procedure, the moving part assigned to node n_k^i is:

$$\hat{x}_k^i = \arg\max_j b(x_k^i = j), \quad (7)$$

where the belief at node n_k^i is defined as:

$$b(x_k^i) = \Phi(x_k^i) \prod_{n\in\mathcal{N}(n_k^i)} m_{n,n_k^i}(x_k^i). \quad (8)$$

In Fig. 7, we give an analysis of the potentials in (1), including the importance term, the spatial compatibility term, and the chronological term. By the comparisons, it can be seen that the importance term retains the important moving parts, the spatial compatibility term assembles moving parts that are compatible with each other in their partition boundaries, and the chronological term ensures that temporally consecutive MRF nodes are assigned consecutive part movements.

4.3 Part Movement Stitching Optimization

Since the moving parts on the same frame of synopsis usually come from different frames of the input video, they may not exactly stitch with each other. Fig. 8a shows an example of such artifact. We compute a 2D vector v_k^i for each moving part at node n_k^i, and shift the part by the vector in the spatial domain, to stitch moving parts together.

The shifting vectors are computed via a part movement stitching optimization whose objective function is formulated as a linear combination of three weighted energy terms:

$$E = \omega_d E_d + \omega_t E_t + \omega_s E_s, \quad (9)$$

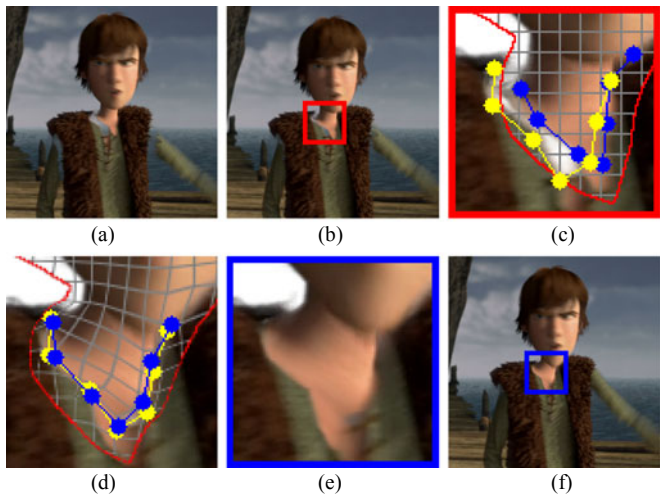


Fig. 8. Part movement stitching optimization. (a) Result after part assembling optimization. The gaps between adjacent moving parts are apparent. (b) The gaps are nearly eliminated after part stitching optimization. (c) The blue boundary do not exactly match with the yellow boundary, thus small seam appear between moving parts. (d) The boundary of head part is deformed to let its boundary exactly match with that of the torso part. (e) The seam between the head part and torso part is eliminated. (f) The final result without gaps.

TABLE 1
The Experimental Data in Object Movements Synopsis Processing

Examples	N	K	M	Time of Part Assembling	Time of Part Stitching
Fig. 9	151	4	60	2s	4s
			48	1s	3s
Fig. 10	500	3	217	7s	NoA
			190	5s	
Fig. 11	180	2	63	2s	3s
Fig. 12	180	3	64	3s	5s
Fig. 13	1353	3	423	10s	NoA
Fig. 15	998	3	520	12s	NoA

For each example, we give the length of input video (N), the number of semantic parts (K), the length of synopsis (M), the time used in part assembling optimization and the part stitching optimization. "NoA" in the "Time of Part Stitching" column means part stitching is not needed.

where ω_d, ω_t and ω_s are weights of the relative terms. Term E_d prevents the moving parts from being shifted too far. It is defined as the sum of all the shifting vectors:

$$E_d = \sum_{i,k} \left\| v_k^i \right\|^2. \tag{10}$$

Term E_t guarantees the temporal coherence of moving parts after shifting, i.e., temporally adjacent moving parts should be shifted by similar vectors. We minimize the change in the second derivative of the vectors over time:

$$E_t = \sum_{i,k} \left\| v_k^{i-1} - 2v_k^i + v_k^{i+1} \right\|^2. \tag{11}$$

Term E_s is developed to reduce gap artifacts between spatially assembled moving parts. By eliminating the gaps, we stitch the parts together.

Thus, E_s measures the distance between adjacent boundary curves:

$$E_s = \sum_{i,k,l,t} \left\| c_k^i(s_t) - c_l^i(s_{t+\Delta}) \right\|^2. \tag{12}$$

where $c_k^i(s_t)$ means a point s_t on curve c_k^i, and t is the index of the point. When the boundaries have the same length, Δ is zero. When the boundaries have different length, Δ is a non-zero offset to match the short boundary with a sub curve of long boundary that has the most similar curvatures.

Objective function (9) has the form of non-linear linear squares (NLLS), which is well-studied. One of the most popular methods to minimize (9) is the Levenberg-Marquardt (LM) algorithm [41], which is a standard iterative technique that can be viewed as a combination of steepest descent and Gauss-Newton. We use the GPL native levmar [42], an implementation of LM algorithm in c/c++, to minimize (9). LM algorithm requires computing the Jacobian of each energy term. However, the Jacobian of the third term E_s is unavailable since it is based on the coordinates of points on curves but not the shifting vectors v_k^i. Instead of computing the Jacobian analytically, we approximate the Jacobian by finite difference method.

To receive desirable results, we minimize (9) twice by setting different weights for the terms. First, we give larger weight to term E_s by setting $\omega_d = 1, \omega_t = 30, \omega_s = 100$. The optimization process effectively reduces the gaps between moving parts. Second, we favor term E_t by setting $\omega_d = 1, \omega_t = 80, \omega_s = 30$. After optimization, the shifting vectors become more temporally coherent.

Fig. 8b shows a result after shifting the moving parts by the vectors computed by part stitching optimization. Compared to Fig. 8a, the gaps between moving parts are almost eliminated. However, since the boundaries do not exactly match with each other (Fig. 8c shows that the blue boundary does not exactly match with the yellow boundary), there are small seams between moving parts. To eliminate the small seams, we move the boundary of a moving part to overlap with the other boundary, and deform the rest of the part by using the method [43]. Fig. 8d shows that the blue and yellow boundaries coincide. We can see from Fig. 8e that the small seam disappears. Fig. 8f gives the final result without gaps.

5 Results and Discussions

The proposed algorithm is developed in C++ and tested on an Intel Core 2 Duo 2.5 GHz computer with 2 GB RAM. In this section, we present a variety of video object movements synopsis results generated using our part-based synopsis method. The objects that can be processed by our approach widely exist in various kinds of videos. Most of the experimental videos are downloaded from YouTube, and the others are extracted from movies. The length of input video varies from a hundred to several thousands of frames, while the synopsis of the input video is relatively much shorter. We partition an object into K semantic parts. For different objects in different videos, K is different, which varies from 2 to 4 in this paper. Table 1 lists the experimental information for each example. For more intuitive viewing of synopsis results in this paper, please refer to the supplemental videos, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TVCG.2013.2297931.

The preparation phase of our algorithm, that is, the object partition stage, is time-consuming, since rotoscoping and matting needs user interaction. We use the effective and robust software "Silhouette" of SilhouetteFX, LLC to help us partition objects. For a skilled user, it takes about five minutes to extract a moderately complex part sequence in 100 frames. Our part movement assembling optimization is efficient, which has a linear complexity with respect to the length of the input video. The part movement stitching optimization is efficient too due to the small number of points sampled on partition boundaries. Table 1 presents the time consumed in the two optimizations of each example.

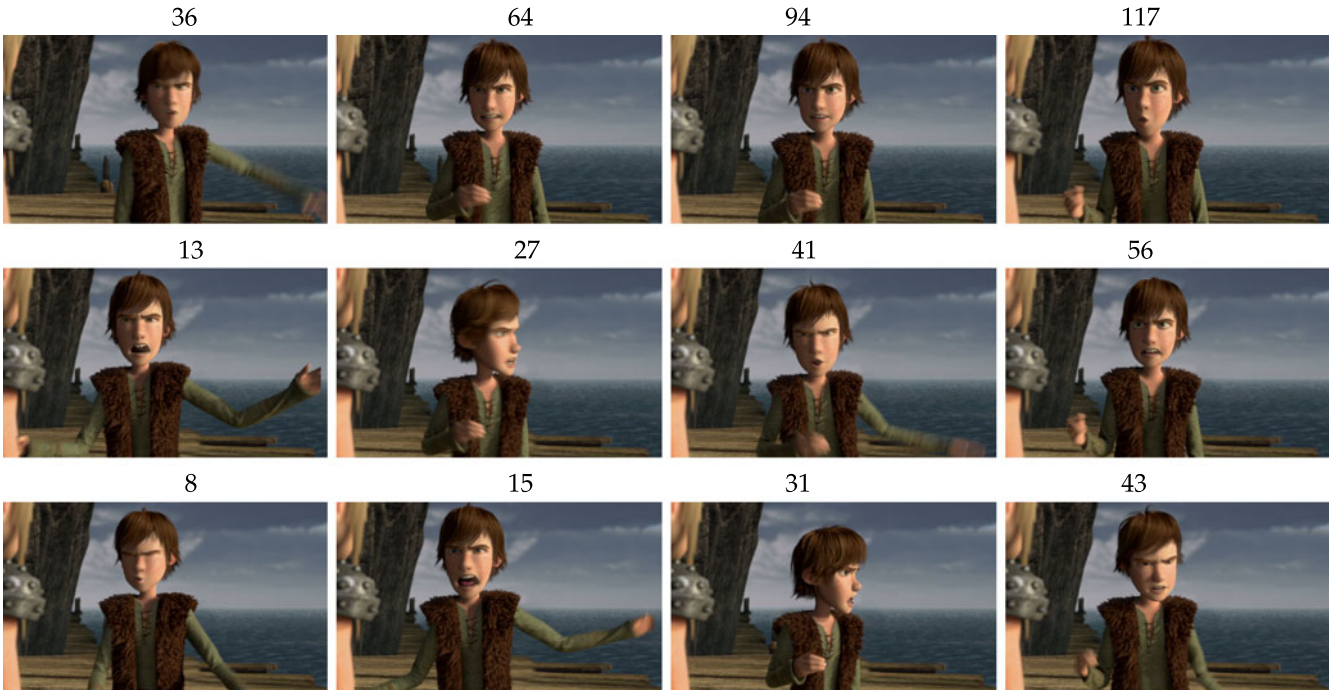


Fig. 9. The synopsis of movie clip "How to Train Your Dragon". The top row shows the most important object movements in four frames of the input video (151 frames), in which the 64th and 94th frames are two repeated actions. The middle row shows four frames of a synopsis with 60 frames. In the bottom row, we further eliminate repeated movements of right arm and obtain a more compact synopsis with only 48 frames.

We do experiments on different types of input videos and movie clips, and compare our method to the typical frame-based and object-based synopsis methods. Our approach is better than other methods in that it can effectively reduce redundancies existing in part movements of object, which cannot be removed by previous approaches. The object-based video synopsis methods simply cannot deal with videos with only one object, while the frame-based methods discard a frame as a whole when a small portion of the frame is moving. By assembling important movements of different object parts on different frames together, the synopsis produced by our method is not only compact, but also preserves most important motion information for fast browsing, especially for online and mobile applications. Our method is a good complement to previous methods. Although there is no single method to effectively condense all kinds of videos, we show in the following experiments that the cooperation of our method and the others (frame-based or object-based methods) can obtain much better results.

In Fig. 9, we generate part-based synopsis for the movie clip "How to Train Your Dragon". In the source video, the boy first moves his right arm, and then his left arm. When the right arm is moving, the left arm is static, and vice versa. Our method effectively selects the most important movements existing in each part, and assembles them together to obtain synopsized object movements. We divide the object into four parts: the head, the torso, the left and the right arms. First, we condense the input video into 60 frames as shown in the middle row of Fig. 9. We find that the video can be further condensed by eliminating repeated movements of right arm. A straightforward method is to detect repeated part movements and remove them from the part candidate set. The bottom row of Fig. 9 shows a more compact synopsis with only 48 frames after removing repeated redundancies existing in the right arm. This example shows that our approach can generate part-based synopsis of movie clips for fast browsing and video editing.

In Fig. 10, we show that our method can be used to synopsize/edit the movements of a car's door opening system. In the input video, the car first opens its right door, then the left door, and lastly, the middle panel. Our method opens the three parts simultaneously which produces a short synopsis. The input video contains 500 frames. After removing inter-part static redundancies among doors and panel, we got a synopsis with only 217 frames, as show in the middle row of Fig. 10. However, since the middle panel opens very slowly, the synopsis is still lengthy. To obtain a more compact synopsis, we set the length of synopsis as 109. Our part assembling operator automatically accelerate the middle panel's moving. The bottom row of Fig. 10 shows four frames of our final synopsis with only 109 frames. Fig. 9 and Fig. 10 demonstrate that our method can remove not only inter-part static redundancies, but also the inner-part redundancies existing in moving parts such as repeated movements or the parts that move slowly.

Fig. 11 shows another synopsis result—Curiosity Mars Rover. The rover first rotates part A (the blue part), and then part B (the red one). In our synopsis, the two parts are assembled to rotate simultaneously. For example, in the second column of Fig. 11, when part A is rotating, part B in the input video doesn't rotate. But part B in the synopsis has rotated 30 degrees which comes from the 87th frame of the input video. The input video has 180 frames, and the synopsis has 63 frames, which accelerates the browsing of Rover movements.

In Fig. 12, the top row shows an input video with blooming flowers. From the 9th to 118th frame, only the left flower

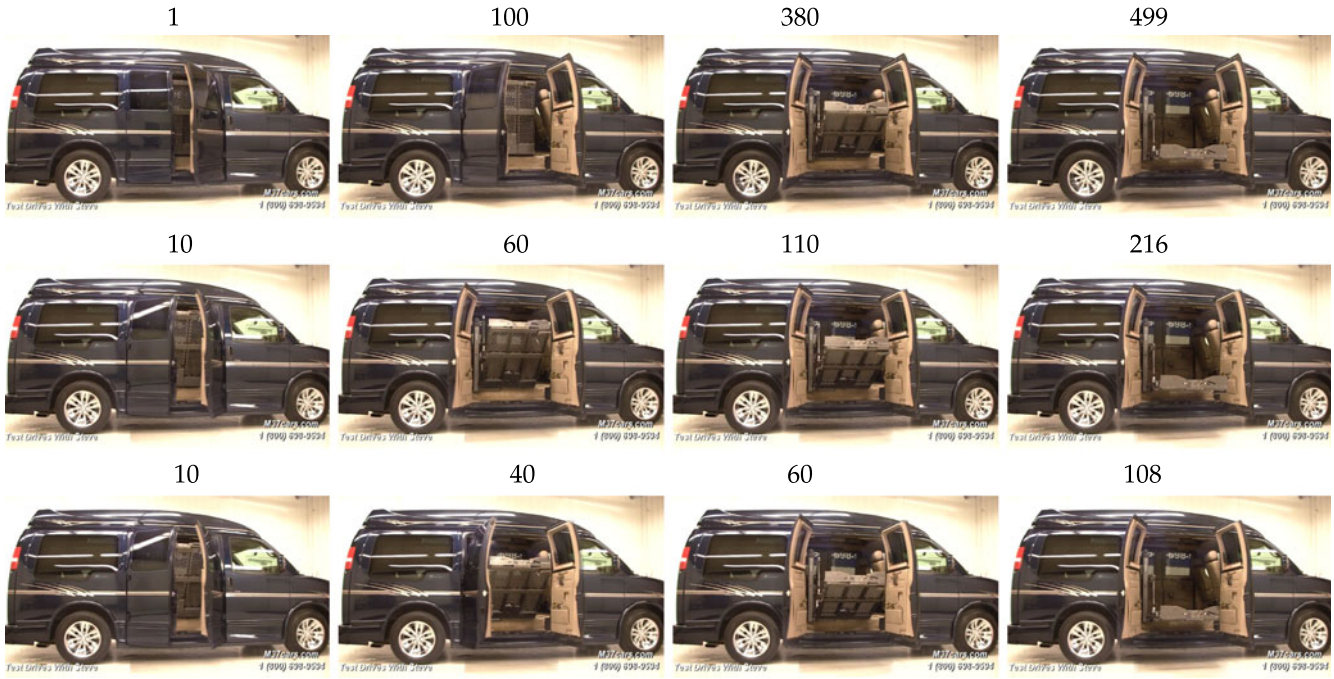


Fig. 10. The synopsis of car door opening system. The top row shows four frames of the input video with 500 frames. The car first open its right door, then the left door, and finally the middle panel. The middle row shows four frames of the synopsis with 217 frames where the three parts move simultaneously. The bottom row shows a more compact synopsis with only 109 frames by accelerating the middle panel.

is blooming. Then the right flower begins to bloom from the 119th frame.The original flower blooming process is lengthy. Although the input video can be compressed into a shorter video by uniform key-frame method [1], the result is not natural. Our method effectively eliminates the time difference of blooming of the two flowers, and make them blooming simultaneously. The second row of Fig. 11 shows our synopsis result. From the 2th fame, both the left and right flowers begin to bloom. The input video has 180 frames, while our synopsis has only 60 frames.

In Fig. 13, we show a robot with two arms picking stuffs on the ground and putting them into a blue plastic bucket. The input video has 1,353 frames, and the first row shows six of them. The input video can be divided into three sections: during the first section from frame 1 to 450, the robot picks two stuffs in the lower left corner, in the second section (frame 451 to 815), the robot carries the stuffs and puts them into the bucket, and in the last section (frame 816 to 1,353), the robot picks another two stuffs in the top right corner. Our method can condense the first and third sections, as in both sections the robot picks a stuff using one arm, and then picks another stuff using other arm. We can effectively condense the movements of the robot by making the robot to pick two stuffs simultaneously, and obtain a good synopsis, as shown in the second row. However, for the second section, we think the key-frame based method can receive better result. We use the key-frame based method [1] to condense the second section. By combining

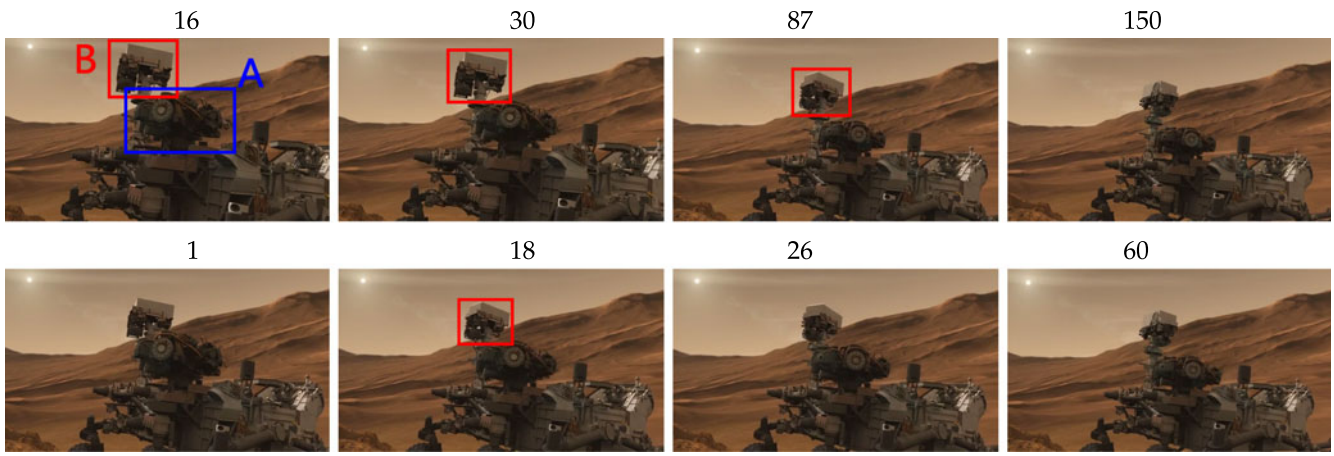


Fig. 11. The synopsis of Curiosity Mars Rover. The top row shows four frames of the input video (180 frames): 16th, 30th, 87th and 150th frames. The rover first rotates part A, then rotates part B. Bottom row shows four frames of our synopsis with only 63 frames which rotates part A and B simultaneously: 1th, 18th, 26th and 60th frames. The second column shows that part B do not move in the input video, but in the synopsis it has rotated 30 degrees which comes from the 87th frame of the input video.

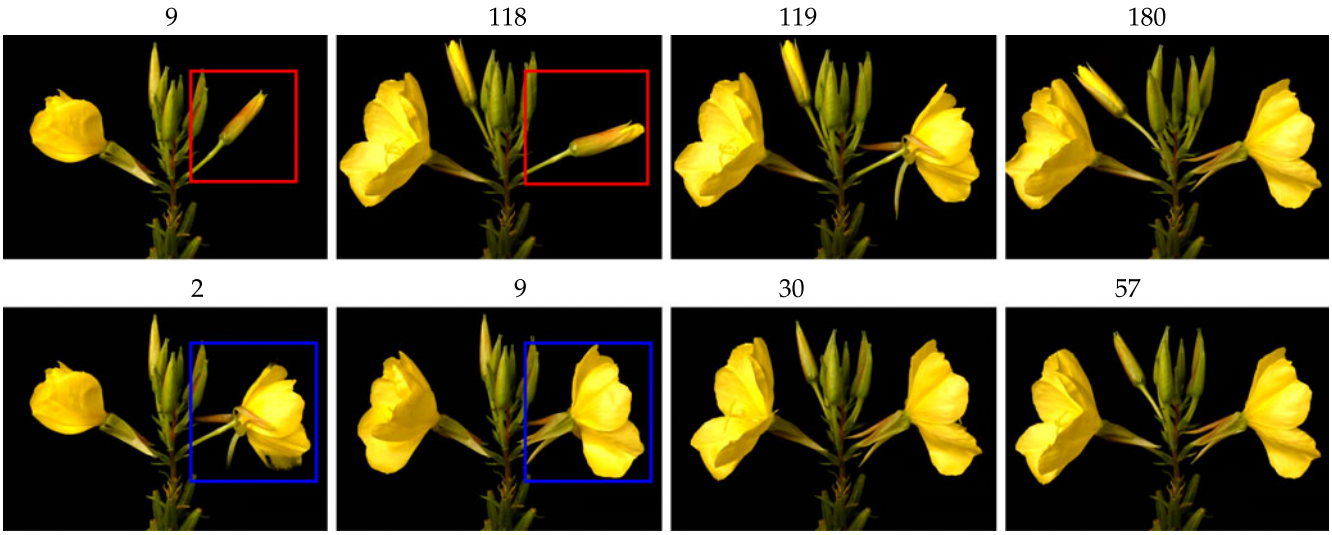


Fig. 12. The top row shows a time-lapsing blooming process of flower (180 frames), produced by uniform key-frame method [1]. The blooming process cannot be condensed by the key-frame based method any more, otherwise it would look unnatural. The red boxes show that the left flower blooms first, then the second flower blooms. The second row shows the result of our synopsis method (64 frames). The blue boxes show that the two flowers bloom together.

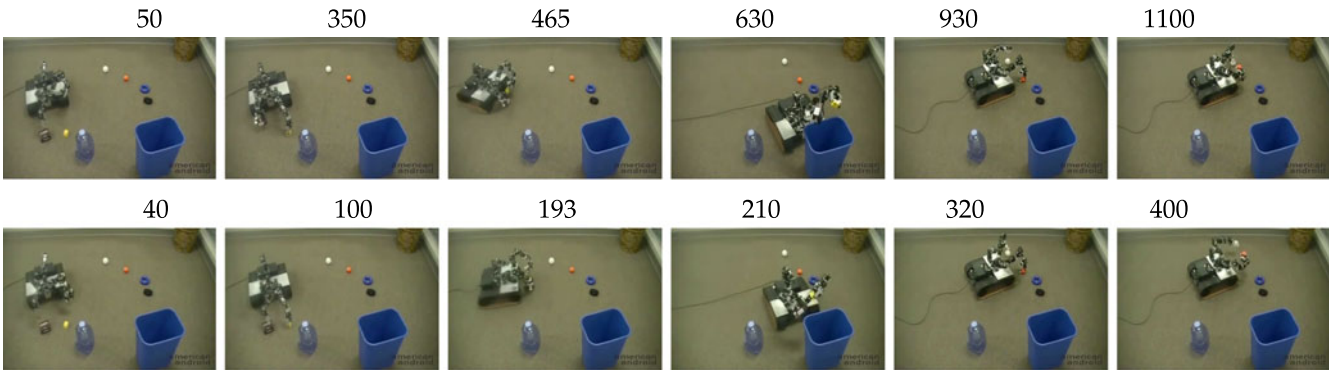


Fig. 13. The top row shows a robot with two arms picking stuffs on the ground and putting them into a blue plastic bucket (1,353 frames). The second row shows a result produced by combining the key-frame method [1] and our method (423 frames).

our method and the key-frame method, the input video can be successfully condensed, and to receive an effective and natural synopsis (423 frames).

In Fig. 14, we show why our method is better than the key-frame method [1] when processing the first and third sections of Fig. 13. The top row shows two consecutive frames of our synopsis, and the second row shows two consecutive frames of the synopsis produced by fast forward method. The first frame of each row has the right arm from the same frame in the input video. The second frame shows that the right arm of the second row moves a much larger step, which is not natural. This is because the key-frame based method simply discard the frames between two key frames, thus many motion details are lost, leading to rushing artifacts. Our method condenses the video by moving parts simultaneously, and keeps important movements of each part.

In Fig. 15, we compare our method with object-based method [2]. The input video shows a common animation conversation between a man and a woman. The man talks first, then the woman responses. We condense the input video with 998 frames to a synopsis with 520 frames. Object-based method [2] shifts the movements of woman forward along the time axis, making the man and woman talking simultaneously which is not realistic. Our method condenses each object separately by reducing redundancies existing in objects themselves, and obtains a synopsis with

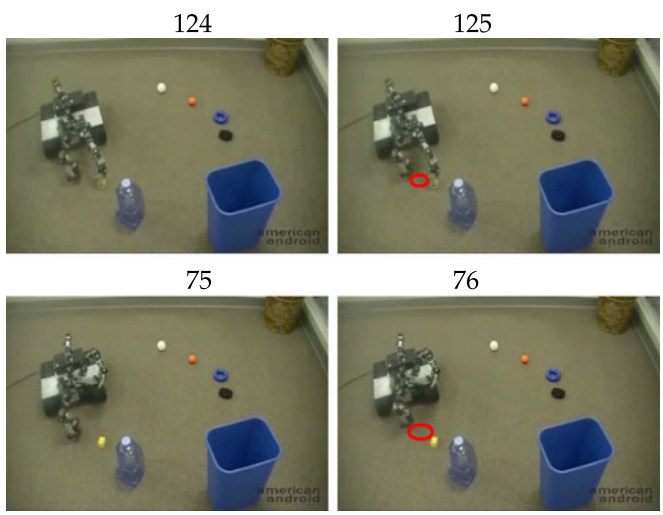


Fig. 14. The first row shows two consecutive frames of our synopsis, and the second row shows two frames of the synopsis produced by fast forward method [1].

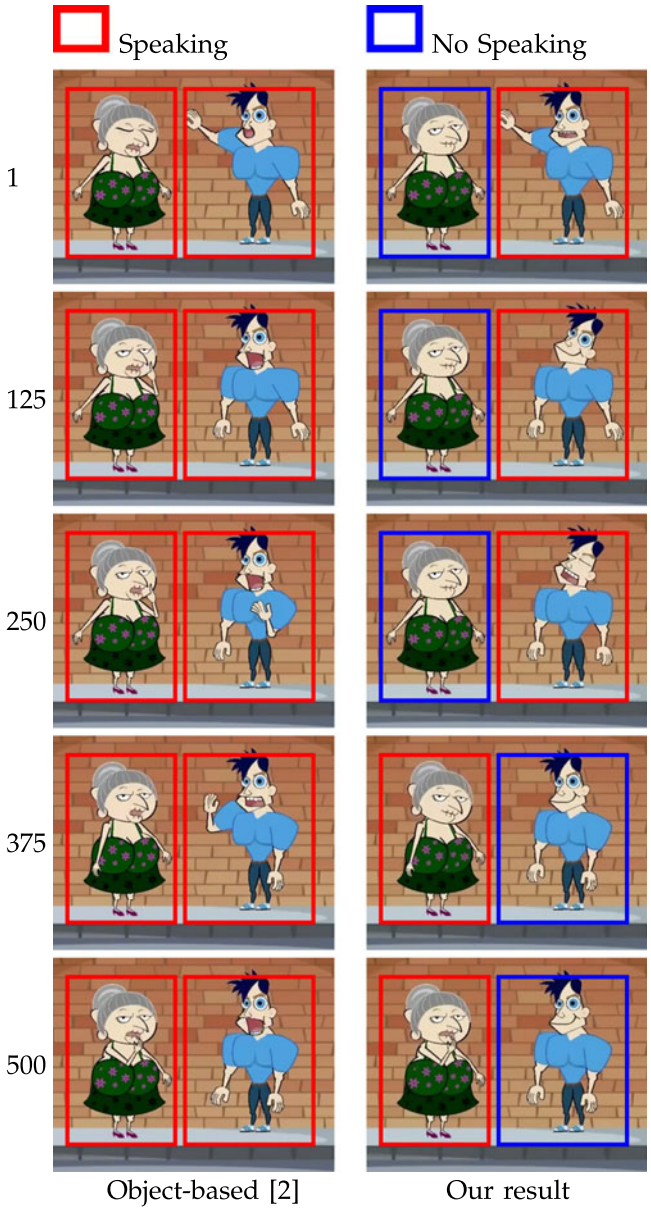


Fig. 15. *Left Column:* The result of object-based method [2]. The man and woman are made to talk simultaneously, which is not realistic. *Right Column:* Our method condenses each object separately to obtain a synopsis with the same length as [2] meanwhile keeping the order of speakers.

the same length. Our method keeps the order of speakers, and the result is more natural.

User Study. We conduct a user study to validate the effectiveness of our part-based object movements synopsis method. We have 30 participants. Each time, a participant was first shown a source video and then two synopsis results of the source video side by side. On the left was the result of frame-based method [1] or object-based method [2], and on the right was our synopsis result. For examples from Figs. 9 to 13, we compared our method with frame-based method. For example in Fig. 15, we compared our method with object-based method. Every participant watched the six examples and each time was asked to answer "Yes" or "No" to the following five questions: (1) Do you think the synopsis on the right is interesting? (2) Do you think the left synopsis is rushing? (3) Do you care about the moving details of object? (4) Do you think the right synopsis represent the source video well? (5) Do you agree with the right synopsis result is better than the left one?

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

$$R = \left(\sum_{i=1}^{30}\sum_{j=1}^{6} A_{ij}\right) \Big/ 180 * 100\%. \tag{13}$$

The "Yes" rates of the five questions were 89, 72, 75, 84, and 72 percent, respectively. From these testing data, we see that most users think our work is more interesting and our results are better than the frame-based or object-based methods.

From the above examples and discussions, we have shown that our method has the following advantages: (1) it can produce more compact and natural synopsis result, as we remove the inter-part and inner-part redundancies but retain part-wise important motions; (2) it is a good complement to previous synopsis methods such as keyframe-based method and object-based method; and (3) it can also be considered as a video editing method; by recombining object parts, we obtain new video synthesis results.

Limitations. Our method is largely based on the assumption that object parts are the basic building-blocks which can be extracted and reorganized for the purpose of removing part-based redundancies. The limitation of our method is damaging the structure of object and introducing new object movements that do not exist in the input video. Thus our method may not work well when new content is not allowed. For example, our method cannot be used to process gesture-talking video, as each gesture has special meaning that must not be destroyed. Although the proposed method's applications are limited by such negative effect, it can still be widely used to synopsize a variety of video clips without modifying the semantic meanings of the input videos.

Second, our method uses matting to partition object. As video matting is a semi-automatic technique, our method requires some user interaction. Although the software "Silhouette" works well for video matting, a skilled user still spends much time on extracting object parts. Static background or background with rich texture may alleviate the burden of user interaction.

Third, it might be challenging for our synopsis method to process objects with large-scale movements such as turning round. The overlaps between parts increase the difficulty of producing perfect synopsis results. However, it can be expected that our method can handle this challenge when condensing 3D object movements rather than those in videos.

6 Conclusion and Future Work

In this paper, based on the observation that part-based redundancies widely exist during the movement process of video object, we present a part-based object movements synopsis method which produces more compact synopsis of video. We partition video object into several semantic

parts. The same number of the most important movements are selected from each part movement sequence by part assembling optimization, which are then assembled together to form synopsized object movements. We then use a part stitching optimization to stitch assembled moving parts seamlessly. We demonstrate that our method can effectively remove part-based redundancies and successfully produces more compact synopsis.

In the future, we would like to build a navigation framework based on our synopsis result, where a user can track into the source video to see the original content whenever they find an interesting moving part in the synopsis. We find that our technique can be used to condense 3D object movements whose parts are easier to track. Thus another work is to extend a version of the proposed method for 3D objects.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their valuable comments and insightful suggestions. This work was partly supported by the National Basic Research Program of China (No. 2012CB725303), the NSFC (No. 61070081 and 41271431), Program for New Century Excellent Talents in University (No. NCET-13-0411), GRF grant (ref. 416311) and UGC direct grant for research (no. 2050454), the Open Project Program of the State Key Lab of CAD&CG (Grant No. A1208), the Luojia Outstanding Young Scholar Program of Wuhan University, and the Academic Award for Excellent Ph.D. Candidates funded by Ministry of Education of China (No. 5052012211001). Chunxia Xiao is the corresponding author.

REFERENCES

[1] B. Truong and S. Venkatesh, "Video Abstraction: A Systematic Review and Classification," *ACM Trans. Multimedia Computing, Comm., and Applications*, vol. 3, no. 1, p. 3, 2007.

[2] Y. Pritch, A. Rav-Acha, and S. Peleg, "Nonchronological Video Synopsis and Indexing," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 1971-1984, Nov. 2008.

[3] A. Rav-Acha, Y. Pritch, and S. Peleg, "Making a Long Video Short: Dynamic Video Synopsis," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 435-441, 2006.

[4] Y. Nie, C. Xiao, H. Sun, and P. Li, "Compact Video Synopsis via Global Spatiotemporal Optimization," *IEEE Trans. Visualization and Computer Graphics*, vol. 19, no. 10, pp. 1664-1676, Oct. 2013.

[5] S. Lu, S. Zhang, J. Wei, S. Hu, and R. Martin, "Time-Line Editing of Objects in Video," *IEEE Trans. Visualization and Computer Graphics*, vol. 19, no. 7, pp. 1218-1227, July 2013.

[6] 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.

[7] C. Gianluigi and S. Raimondo, "An Innovative Algorithm for Key Frame Extraction in Video Summarization," *J. Real-Time Image Processing*, vol. 1, no. 1, pp. 69-88, 2006.

[8] Y. Ma, L. Lu, H. Zhang, and M. Li, "A User Attention Model for Video Summarization," *Proc. 10th ACM Int'l Conf. Multimedia*, pp. 533-542, 2002.

[9] C. Taskiran, Z. Pizlo, A. Amir, D. Ponceleon, and E. Delp, "Automated Video Program Summarization Using Speech Transcripts," *IEEE Trans. Multimedia*, vol. 8, no. 4, pp. 775-791, Aug. 2006.

[10] J. Wu, M. Kankanhalli, J. Lim, and D. Hong, *Perspectives on Contentbased Multimedia Systems*, vol. 9, Springer, 2000.

[11] J. Ouyang, J. Li, and Y. Zhang, "Replay Boundary Detection in Mpeg Compressed Video," *Proc. Int'l Conf. Machine Learning and Cybernetics*, vol. 5, pp. 2800-2804, 2003.

[12] A. Schödl and I.A. Essa, "Controlled Animation of Video Sprites," *Proc. ACM SIGGRAPH/Eurographics Symp. Computer Animation*, pp. 121-127, 2002.

[13] H. Kang, X. Chen, Y. Matsushita, and X. Tang, "Space-Time Video Montage," *Proc. IEEE Computer Soc. Conf. Computer Vision and Pattern Recognition*, vol. 2, pp. 1331-1338, 2006.

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

[15] T. Chen, A. Lu, and S.-M. Hu, "Visual Storylines: Semantic Visualization of Movie Sequence," *Computers & Graphics*, vol. 36, no. 4, pp. 241-249, 2012.

[16] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani, "Summarizing Visual Data Using Bidirectional Similarity," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1-8, 2008.

[17] C. Xiao, M. Liu, N. Yongwei, and Z. Dong, "Fast Exact Nearest Patch Matching for Patch-Based Image Editing and Processing," *IEEE Trans. Visualization and Computer Graphics*, vol. 17, no. 8, pp. 1122-1134, Aug. 2011.

[18] Y. Nie, Q. Zhang, R. Wang, and C. Xiao, "Video Retargeting Combining Warping and Summarizing Optimization," *The Visual Computer*, vol. 29, pp. 785-794, 2013.

[19] C. Barnes, D. Goldman, E. Shechtman, and A. Finkelstein, "Video Tapestries with Continuous Temporal Zoom," *ACM Trans. Graphics*, vol. 29, no. 4, p. 89, 2010.

[20] L. Teodosio and W. Bender, "Salient Video Stills: Content and Context Preserved," *Proc. First ACM Int'l Conf. Multimedia*, pp. 39-46, 1993.

[21] Y. Caspi, A. Axelrod, Y. Matsushita, and A. Gamliel, "Dynamic Stills and Clip Trailers," *The Visual Computer*, vol. 22, no. 9, pp. 642-652, 2006.

[22] C. Correa and K. Ma, "Dynamic Video Narratives," *ACM Trans. Graphics*, vol. 29, no. 4, p. 88, 2010.

[23] A. Agarwala, K. Zheng, C. Pal, M. Agrawala, M. Cohen, B. Curless, D. Salesin, and R. Szeliski, "Panoramic Video Textures," *ACM Trans. Graphics*, vol. 24, no. 3, pp. 821-827, 2005.

[24] A. Rav-Acha, Y. Pritch, D. Lischinski, and S. Peleg, "Dynamosaics: Video Mosaics with Non-Chronological Time," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 58-65, 2005.

[25] J. Assa, Y. Caspi, and D. Cohen-Or, "Action Synopsis: Pose Selection and Illustration," *ACM Trans. Graphics*, vol. 24, no. 3, pp. 667-676, 2005.

[26] C. Shen, H. Fu, K. Chen, and S. Hu, "Structure Recovery by Part Assembly," *ACM Trans. Graphics*, vol. 31, no. 6, p. 180, 2012.

[27] J. Wang, P. Bhat, R. Colburn, M. Agrawala, and M. Cohen, "Interactive Video Cutout," *ACM Trans. Graphics*, vol. 24, no. 3, pp. 585-594, 2005.

[28] X. Bai, J. Wang, D. Simons, and G. Sapiro, "Video Snapcut: Robust Video Object Cutout Using Localized Classifiers," *ACM Trans. Graphics*, vol. 28, no. 3, p. 70, 2009.

[29] L. Zhang, H. Huang, and H. Fu, "Excol: An Extract-and-Complete Layering Approach to Cartoon Animation Reusing," *IEEE Trans. Visualization and Computer Graphics*, vol. 18, no. 7, pp. 1156-1169, July 2012.

[30] K. Xu, J. Wang, X. Tong, S.-M. Hu, and B. Guo, "Edit Propagation on Bidirectional Texture Functions," *Computer Graphics Forum*, vol. 28, no. 7, pp. 1871-1877, 2009.

[31] K. Xu, Y. Li, T. Ju, S.-M. Hu, and T.-Q. Liu, "Efficient Affinity-Based Edit Propagation Using Kd Tree," *ACM Trans. Graphics*, vol. 28, no. 5, p. 118, 2009.

[32] C. Xiao, Y. Nie, and F. Tang, "Efficient Edit Propagation Using Hierarchical Data Structure," *IEEE Trans. Visualization and Computer Graphics*, vol. 17, no. 8, pp. 1135-1147, Aug. 2011.

[33] S.-M. Hu, K. Xu, L.-Q. Ma, B. Liu, B.-Y. Jiang, and J. Wang, "Inverse Image Editing: Recovering a Semantic Editing History from a Before-and-After Image Pair," *ACM Trans. Graphics*, vol. 32, no. 6, p. 194, 2013.

[34] J. Wang and M. Cohen, *Image and Video Matting: A Survey*, Now Pub, vol. 3, no. 2, 2008.

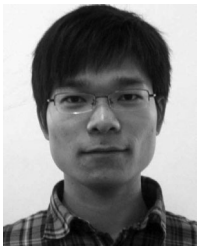
[35] A. Agarwala, A. Hertzmann, D. Salesin, and S. Seitz, "Keyframe-Based Tracking for Rotoscoping and Animation," *ACM Trans. Graphics*, vol. 23, no. 3, pp. 584-591, 2004.

[36] K. He, C. Rhemann, C. Rother, X. Tang, and J. Sun, "A Global Sampling Method for Alpha Matting," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR '11)*, pp. 2049-2056, 2011.

[37] J. Sun, L. Yuan, J. Jia, and H.-Y. Shum, "Image Completion with Structure Propagation," *ACM Trans. Graphics*, vol. 24, no. 3, pp. 861-868, 2005.

[38] R.W. Zhou, C. Quek, and G.S. Ng, "A Novel Single-Pass Thinning Algorithm and an Effective Set of Performance Criteria," *Pattern Recognition Letters*, vol. 16, no. 12, pp. 1267-1275, 1995.

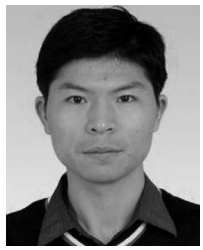
[39] D. Sun, S. Roth, and M. Black, "Secrets of Optical Flow Estimation and their Principles," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR '10)*, pp. 2432-2439, 2010.
[40] J. Yedidia, W. Freeman, and Y. Weiss, "Understanding Belief Propagation and Its Generalizations," *Exploring Artificial Intelligence in the New Millennium*, vol. 8, pp. 236-269, Morgan Kaufmann, 2003.
[41] K. Madsen, H. Bruun, and O. Tingleff, "Methods for Non-Linear Least Squares Problems," 1999.
[42] M. Lourakis, "Levmar: Levenberg-Marquardt Nonlinear Least Squares Algorithms in C/C++," http://www.ics.forth.gr/~lourakis/levmar, 2004.
[43] S. Schaefer, T. McPhail, and J. Warren, "Image Deformation Using Moving Least Squares," *ACM Trans. Graphics*, vol. 25, no. 3, pp. 533-540, 2006.



Yongwei Nie received the BS degree from the School of Computer, Wuhan University, in 2009. He is currently working toward the PHD degree at the School of Computer, Wuhan University, China. His research interests include image and video editing, and computational photography.

Hanqiu Sun received the MS degree in electrical engineering from the University of British Columbia, and the PhD degree in computer science from the University of Alberta, Canada. She is an associate professor at the Chinese University of Hong Kong. Her research interests include virtual reality, interactive graphics/animation, real-time hypermedia, virtual surgery, mobile image/video synopsis and navigation, touch-enhanced simulations.

Ping Li received the BEng degree in computer science and technology from China Jinan University, and the PhD degree in computer science and engineering from the Chinese University of Hong Kong. He is a lecturer at the Hong Kong Institute of Education. His research interests include image/video processing and creative media, including retexturing, stylization, colorization, video summarization, and GPU acceleration.



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, and during February 2012 to February 2013, he visited University of California-Davis for one year. His main research interests include image and video processing, digital geometry processing, and computational photography. Chunxia Xiao is the corresponding author.

Kwan-Liu Ma received the PhD degree in computer science from the University of Utah in 1993. He is a professor of computer science and the chair of the Graduate Group in Computer Science (GGCS) at the University of California, Davis. He leads the VIDi research group and directs the UC Davis Center for Visualization. He was a recipient of the PECASE Award in 2000. His research interests include visualization, high-performance computing, computer graphics, and user interface design. He was a paper chair of the IEEE Visualization Conference in 2008 and 2009, and an associate editor of IEEE TVCG (2007-2011). He is a founding member of the IEEE Pacific Visualization Symposium and the IEEE Symposium on Large Data Analysis and Visualization. He currently serves on the editorial boards of the IEEE CG&A, the *Journal of Computational Science and Discoveries*, and the *Journal of Visualization*. He is a fellow of the IEEE.