Spatio-Temporal Video Segmentation of Static Scenes and Its Applications

Hanqing Jiang, *Graduate Student Member, IEEE*, Guofeng Zhang, *Member, IEEE*, Huiyan Wang, *Member, IEEE*, and Hujun Bao, *Associate Member, IEEE*

***Abstract*—Extracting spatio-temporally consistent segments from a video sequence is a challenging problem due to the complexity of color, motion and occlusions. Most existing spatio-temporal segmentation approaches have inherent difficulties in handling large displacement with significant occlusions. This paper presents a novel framework for spatio-temporal segmentation. With the estimated depth data beforehand by a multi-view stereo technique, we project the pixels to other frames for collecting the boundary and segmentation statistics in a video, and incorporate them into the segmentation energy for spatio-temporal optimization. In order to effectively solve this problem, we introduce an iterative optimization scheme by first initializing segmentation maps for each frame independently, and then link the correspondences among different frames and iteratively refine them with the collected statistics, so that a set of spatio-temporally consistent volume segments are finally achieved. The effectiveness and usefulness of our automatic framework are demonstrated via its applications for 3D reconstruction, video editing and semantic segmentation on a variety of challenging video examples.**

***Index Terms*—3D reconstruction, spatio-temporal segmentation, video editing.**

I. Introduction

WITH the increasing prevalence of digital cameras and intelligent mobile phones, more and more videos are shared and broadcasted over the Internet. An effective video editing tool is highly on demand for users to conveniently modify and enhance the video contents. However, compared to image editing, video editing is much more challenging due to much larger data and difficulty of maintaining the temporal coherence. To thoroughly solve these problems, we need a powerful tool to segment the video into a set of temporally consistent layers. However, spatio-temporal video segmentation is very challenging due to the large number of unknowns, possible colors and motion ambiguities, and complicated geometric structure of the captured scenes.

Fortunately, with the recent advances in 3D vision area [1]–[4] and the increasing prevalence of range sensors (e.g. time-of-flight cameras and Kinects), achieving high-quality depth maps or 3D models becomes much easier now. In the future, most captured images/videos will have depth data. How to appropriately utilize depth information for video segmentation is becoming an important issue. So far there are not many works done for video segmentation utilizing depth information.

In this paper, we propose a novel depth-based video segmentation method which can be used for large-scale 3D reconstruction and many other applications. The objective of our video segmentation method is that the extracted segments not only preserve object boundaries but also maintain the temporal consistency in different images. With the spatio-temporal segmentation results, we can reconstruct the 3D geometry model for a large-scale scene. The spatio-temporal segmentation results can also facilitate many other video applications, such as video editing/stylization and semantic segmentation.

II. Related Work

A. Image/Video Segmentation

During the past decades, many image segmentation methods have been proposed, such as normalized cuts [5], mean shift [6], segmentation via lossy compression [7], and segmentation by weighted aggregation (SWA) [8]. For a video sequence, if we directly use these image-based segmentation methods to segment each frame independently, the segmentation results will be inconsistent for different images due to the lack of necessary temporal coherence constraints.

Some spatio-temporal segmentation methods [9] have been proposed to extend segmentation from single image to video. Two main types of segmentation criteria (i.e. motion and color/texture) were generally used alone or in combination for video segmentation. Motion-based segmentation methods [10] aimed to group pixels which undergo similar motion, and separate them into multiple layers. Many of them [11]–[13] needed to estimate optical flow first, and then segment the pixels based on the learned motion models. Some of them [10], [14], [15] combined motion estimation and segmentation together, and iteratively refined them. However, pure motion-based methods

Manuscript received March 01, 2014; revised August 19, 2014; accepted October 16, 2014. Date of publication November 07, 2014; date of current version December 23, 2014. This work was supported in part by the National Key Technology Research and Development Program of the Ministry of Science and Technology of China under Grant 2014BAK14B01, the NSF of China under Grant 61232011, Grant 61272048, and Grant 61472362, the Specialized Research Fund for the Doctoral Program of Higher Education of China under Grant 20110101130011, Foundation for the Author of National Excellent Doctoral Dissertation of PR China under Grant 201245, and the Zhejiang Province Science and Technology Plan Project under Grant 2014C33070. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Shin'ichi Satoh. *(Corresponding author: Guofeng Zhang.)*

H.-Q. Jiang, G.-F. Zhang, and H.-J. Bao are with the State Key Lab of CAD and CG, Zhejiang University, Hangzhou 310058, China (e-mail: jianghq@cad.zju.edu.cn; zhangguofeng@cad.zju.edu.cn; bao@cad.zju.edu.cn).

H.-Y. Wang is with the School of Computer Science and Information Engineering, Zhejiang Gongshang University, Hangzhou 310018, China (e-mail: cederic@zjgsu.edu.cn).

This paper has supplementary downloadable multimedia material available at http://ieeexplore.ieee.org provided by the authors. This includes three videos that show results of segmentation methods. This material is 71.4 MB in size.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2014.2368273

are difficult to achieve high-quality segmentation results and usually produce inaccurate object boundaries due to the motion ambiguity and the difficulty of accurate optical flow estimation. Some works combined color and motion cues for spatio-temporal segmentation. Khan and Shah [13] proposed a MAP framework for video segmentation combining multiple cues including spatial location, color and motion.

For video segmentation, both spatial and temporal dimensions should be considered. Most approaches handled these two types of dimensions separately. For example, many approaches [16], [17] first performed spatial segmentation of each frame, and then performed temporal grouping to obtain spatio-temporal volumes. Due to the complexity of color, motion and occlusions in a video, it is challenging for spatial segmentation to produce very consistent segments in different images, so the obtained spatio-temporal segments by temporal grouping will easily contain obvious artifacts. Some methods [18]–[22] employed an online scheme to obtain consistent segments across frames, that each frame was segmented according to the segmentation information propagated from previous frames. Zitnick *et al.* [15] proposed to combine segmentation and optical flow estimation together to produce consistent segments for a pair of images. Vázquez-Reina *et al.* [23] proposed to extract multiple super-pixel flow trajectories as segmentation hypotheses and temporally segment the video by the competition of the trajectories. Galasso *et al.* [24] proposed a novel spectral clustering framework by incorporating motion based superpixel affinities. Chang *et al.* [22] developed a temporal superpixel video representation by modeling temporal flow through bilateral Gaussian process. However, it is difficult for all these methods to handle significant occlusions, where large groups of segments appear or disappear. Our method uses a matching graph to initialize volume segments, which is similar to the super-pixel flow hypothesis in [23]. Different from [23], our method achieves more consistent spatio-temporal volume segmentation in an iterative optimization way, and better handles the occlusion problem by incorporating 3D depth information. Recently, Abramov *et al.* [25] proposed a real-time spatio-temporal segmentation method for color/depth videos captured by a Kinect, which is quite similar to our depth-based scheme. However, [25] only used depths to improve interaction weights and temporal consistency was enhanced by optical flow, while our segmentation uses depth information for collecting multi-view statistics to ensure temporal consistency.

Some space-time segmentation methods [11], [12] were proposed to combine spatial and temporal grouping together, by treating the image sequence as a 3D space-time volume and attempting a segmentation of pixel volumes. These methods typically constructed a weighted graph by taking each pixel as a node and connecting the pixels in the spatio-temporal neighborhood of each other. Normalized cuts was typically used to partition the spatio-temporal volume. Some space-time segmentation methods defined a high dimensional feature vector for each pixel by integrating multiple cues (such as color, space, motion, and time), and clustered these feature points via mean shift analysis [26], [27], GMM [28], or hybrid strategy [29]. Grundmann *et al.* [30] proposed an efficient hierarchical graph-based spatio-temporal segmentation over 3D video volume, and Xu *et al.* [31] extended it to a streaming framework. Lezama *et al.* [32] extended [30] by incorporating long-range motion cues with occlusion reasoning. However, these space-time methods construct a volume representation on the entire video, which inevitably costs huge memory for a long sequence. Besides, all these methods are sensitive to large displacement with significant occlusions. Especially if an object temporarily disappears due to occlusion or out-of-view, it is quite challenging for these methods to cluster the corresponding regions into the same segment.

Our work is also closely related to joint segmentation techniques [33], [34], which simultaneously segmented the reconstructed 3D points and the registered 2D images. Given the multiple view images, they aimed to semantically organize the recovered 3D points and obtain semantic object segmentation, which required user assistance. In contrast, our method can automatically obtain a set of spatio-temporally consistent volume segments from a video sequence.

Gallup *et al.* [35] proposed to use temporal video segmentation for reconstruction of more general scene containing non-planar structures, which is similar to our method. However, this method requires complicated learning for segmentation of piecewise planar and non-planar regions. In comparison, our method can achieve highly consistent spatio-temporal video segmentation without any learning priors. By utilizing the depth redundancy in multiple frames, our spatio-temporal segmentation is rather robust to occlusions and out-of-view. More importantly, our reconstructed geometry models have 3D segmentation labeling and can be used for many other applications such as 3D/video editing, which may not be achieved by previous 3D reconstruction methods.

In summary, spatio-temporal segmentation is still a very challenging problem. Previous approaches generally have difficulties in handling large displacement with significant occlusions. In this paper, we show that by associating multiple frames on the inferred dense depth maps, surprisingly spatio-temporal consistent segments can be obtained from video sequences. The high-quality segmentation results can benefit many other applications, such as 3D reconstruction, video editing, and non-photorealistic rendering.

B. Video Editing

With the increasing prevalence of digital video cameras, video editing has been steadily gaining in importance. Many video editing techniques have been developed during the past decades. The main difficulty of video editing is how to maintain the temporal coherence among temporally neighboring frames. Sand and Teller [36] proposed to spatio-temporally align one video with another with similar camera trajectories for performing some video editing operations, such as background subtraction and video composition. Several techniques have been proposed to treat video as a space-time volume data, which successfully demonstrated its capability for video stylization [37], segmentation [38], completion [39] and summarization [40]. However, these techniques are typically limited to the videos captured by stationary cameras. For handling more complex cases (e.g., the camera can freely move), we generally need to recover depth/3D, motion and even layer information. With the recent advances of structure-from-motion and multi-view stereo techniques, a few techniques have been proposed to perform video editing based on high-quality depth information. Xiao *et al.* [41] proposed a video composition technique which can extract a static 3D object from a sequence and seamlessly insert it to another sequence. Bhat *et al.* [42] proposed to use some high resolution photos to enhance a video of a static scene, based on multiview stereo and image-based rendering techniques. Zhang *et al.* [43] presented a very impressive video editing system based on dense depth recovery and layer separation, which can create various kinds of refilming effects from one or multiple input videos. In general, these methods rely on the correspondences computed by depth or motion information to maintain the temporal coherence, and may have accumulation error or drift problem for a long sequence. In comparison, with spatio-temporal segmentation and 3D representation, we can directly perform video editing in a 3D way, so that the operations are much simpler than those in a 2D way, and the temporal coherence is maintained more easily.

III. Our Approach

Given a video sequence of n frames, our objective is to estimate a set of spatio-temporal volume segments $S = \{S^k | k = 1, 2, \ldots, K\}$, and fuse the volume segments to reconstruct a complete 3D scene, where K is the volume segment number. For a pixel $\mathbf{x}_t$ in frame t, we denote $S(\mathbf{x}_t) = S^k$ if $\mathbf{x}_t \in S^k$. The color of pixel $\mathbf{x}_t$ is denoted as $I(\mathbf{x}_t)$, defined in Luv color space. Denoting by $z_{\mathbf{x}_t}$ the depth value of pixel $\mathbf{x}_t$, the disparity $D(\mathbf{x}_t)$ is defined as $D(\mathbf{x}_t) = 1/z_{\mathbf{x}_t}$ by convention.

Our system overview is shown in Fig. 1. We assume that a depth map is available for each frame of the input video. The depth data could be got by using a depth camera or multi-view stereo techniques [2], [3], [44]. In our experiments, we start by using the structure-from-motion (SFM) method proposed in [45] to recover the camera motion parameters from the input video sequence. The set of camera parameters for frame t is denoted as $\mathbf{C}_t = \{\mathbf{K}_t, \mathbf{R}_t, \mathbf{T}_t\}$, where $\mathbf{K}_t$ is the intrinsic matrix, $\mathbf{R}_t$ is the rotation matrix, and $\mathbf{T}_t$ is the translation vector. With the recovered camera poses, we then employ the multi-view stereo method of Zhang *et al.* [3] to recover a set of consistent depth maps. With the computed depth maps, we first perform spatial segmentation for each frame with probabilistic boundary, and then iteratively optimize the segmentation results by enforcing the temporal coherence constraints among multiple temporal frames. The spatio-temporal segmentation can be used for many applications such as 3D reconstruction, video editing, stylization and semantic segmentation.

IV. Spatial Segmentation with Probabilistic Boundary

Directly obtaining spatio-temporal volume segments in a video is difficult due to the large number of unknowns and the possible geometric and motion ambiguities in the segmentation. Therefore, we design an iterative optimization scheme to achieve spatio-temporal video segmentation. For initialization, instead of directly segmenting each frame independently, we first compute the probabilistic boundary map by collecting the statistics of segment boundaries among multiple frames. Then we perform spatial segmentation for each frame independently with the computed probabilistic boundary maps. Our experimental results demonstrate that much more consistent segmentation results can be obtained than those of directly using mean shift algorithm.

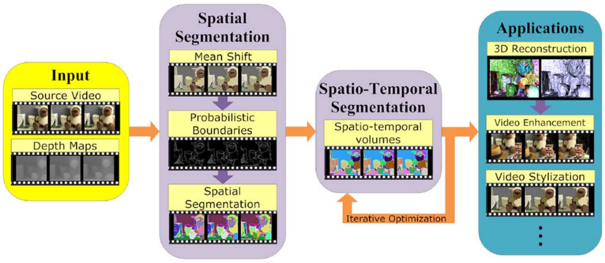


Fig. 1. System overview.

A. Probabilistic Boundary

We first use mean shift algorithm [6] to segment each frame independently with the same parameters. The 2D segments in frame t are denoted as $s_t = \{s_t^k | k = 1, \ldots, K_t^0\}$, where K_t^0 denotes the number of segments in frame t produced by mean shift. For a pixel $\mathbf{x}_t$ in frame t, we denote $s(\mathbf{x}_t) = s_t^k$ if $\mathbf{x}_t \in s_t^k$. Fig. 2(b) shows the segmentation results of the selected frames, which are not consistent in different images. The segmented boundaries are quite flickering, and a segment may span over multiple layers, which is obviously not good enough as a starting point for spatio-temporal segmentation.

With the computed depths, we can project each pixel to other frames to find the correspondences. Considering a pixel $\mathbf{x}_t$ in frame t, with the estimated depth value $z_{\mathbf{x}_t}$, its projection $\mathbf{x}_{t'}$ in frame t' can be computed as follows:

$$\mathbf{x}_{t'}^h \sim z_{\mathbf{x}_t} \mathbf{K}_{t'} \mathbf{R}_{t'}^\top \mathbf{R}_t \mathbf{K}_t^{-1} \mathbf{x}_t^h + \mathbf{K}_{t'} \mathbf{R}_{t'}^\top (\mathbf{T}_t - \mathbf{T}_{t'}) \qquad (1)$$

where the superscript h denotes the vector in the homogeneous coordinate system. The 2D point $\mathbf{x}_{t'}$ is computed by dividing $\mathbf{x}_{t'}^h$ by the third homogeneous coordinate. Then we compute the probabilistic boundary as follows:

$$p_b(\mathbf{x}_t, \mathbf{y}_t) = \frac{1}{n_v} \sum_{t'} [s(\mathbf{x}_{t'}) \neq s(\mathbf{y}_{t'})] \qquad (2)$$

where $\mathbf{y}_t$ is a neighboring pixel of $\mathbf{x}_t$ in frame t, and n_v denotes the number of valid mapping. A mapping is defined to be valid, if the projection points $\mathbf{x}_{t'}$ and $\mathbf{y}_{t'}$ in frame t' are neither occluded nor out-of-view. If $p_b(\mathbf{x}_t, \mathbf{y}_t)$ is large, it is very likely that there is a boundary across pixels $\mathbf{x}_t$ and $\mathbf{y}_t$. Compared to the traditional segmentation boundaries in a single image, our probabilistic boundary map is computed with multiple frames, which is robust to image noise and occasional segmentation errors. The computed probabilistic boundary maps are shown in Fig. 2(c), which are surprisingly consistent among different frames. The reason is that mean shift segmentation can preserve object boundaries well. Although the generated segment boundaries by mean shift may be occasionally inaccurate in one frame, it still has large chance to be accurate in other frames. By collecting the boundary statistics in multiple frames, the computed probabilistic boundaries can naturally preserve the object boundaries and maintain consistency in neighboring frames.

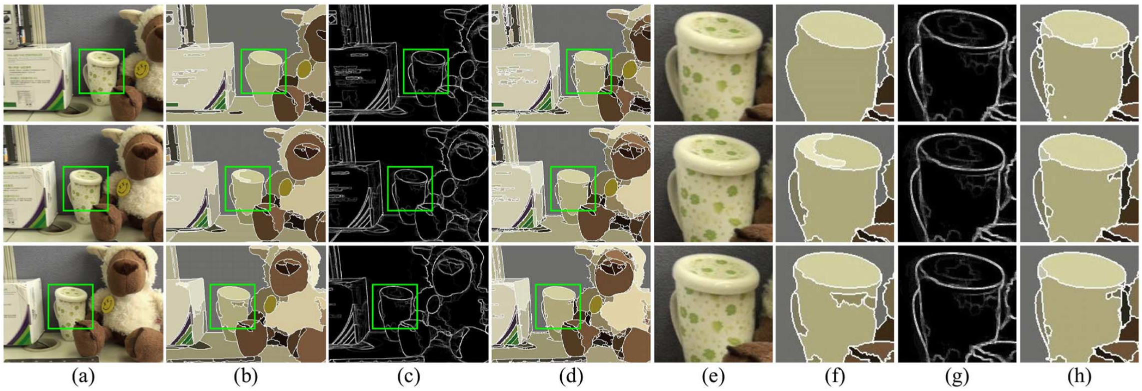


Fig. 2. Spatial segmentation with probabilistic boundary. (a) Three selected frames. (b) The segmentation results with mean shift. (c) The computed probabilistic boundary maps. (d) Our spatial segmentation results. (e)–(h) The magnified regions of (a)–(d). Compared to the results of mean shift, our segmentation results better preserve object boundaries and are much more consistent in different images.

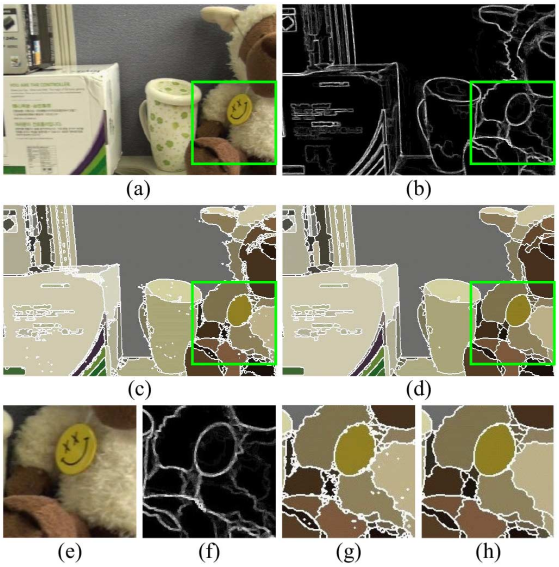


Fig. 3. Flow illustration of spatial segmentation. (a) One original image. (b) The computed probabilistic map. (c) The segmentation results by watershed algorithm based on the computed probabilistic map. (d) After solving (3), the unlabeled pixels are fused into nearby segments. (e)–(h) The magnified regions of (a)–(d), respectively.

B. Spatial Segmentation

With the computed probabilistic boundary map, we use the watershed algorithm [46] to segment the image. We compute a topographic surface $\mathcal{T}(\mathbf{x}_t) = \max_{\mathbf{y}_t \in N(\mathbf{x}_t)} p_b(\mathbf{x}_t, \mathbf{y}_t)$, the maximal probabilistic boundary over the 4 connected probabilistic edges for each pixel, and apply watershed transformation on the surface. The topological map is clipped with a threshold value δ to avoid over segmentation. Fig. 3(c) shows the segmentation result. We notice that some quite small segments appear around the areas with strong probabilistic boundaries, most of which are segmentation noise and do not consistently appear in neighboring frames. So, we eliminate segments that are too small (with less than 30 pixels), and set the pixels in these segments as unlabeled ones. The remaining 2D segments in frame t are denoted as $s_t = \{s_t^k | k = 1, \ldots, K_t\}$. The set of unlabeled pixels is denoted as Φ_t, which will be assigned to these K_t segments. We use $s(\mathbf{x}_t)$ to denote the assigned 2D segment for pixel $\mathbf{x}_t$.

For each frame t, we define the following energy for spatial segmentation:

$$E(s_t) = \sum_{\mathbf{x}_t \in \Phi_t} \Big(E_d\big(s(\mathbf{x}_t)\big) + \sum_{\mathbf{y}_t \in N(\mathbf{x}_t)} E_s\big(s(\mathbf{x}_t), s(\mathbf{y}_t)\big) \Big) \tag{3}$$

where $N(\mathbf{x}_t)$ denotes the set of neighbors of pixel $\mathbf{x}_t$. Data term E_d measures how well the pixels fit the assigned K_t clusters, and the spatial smoothness term E_s encodes the segmentation continuity.

The data term E_d is defined using the Gaussian models of color, disparity, and spatial distributions

$$\begin{aligned} E_d(s(\mathbf{x}_t)) = & -w_c \log \mathcal{N}(I(\mathbf{x}_t) | \mu^c_{s(\mathbf{x}_t)}, \Sigma^c_{s(\mathbf{x}_t)}) \\ & - w_d \log \mathcal{N}(D(\mathbf{x}_t) | \mu^d_{s(\mathbf{x}_t)}, \Sigma^d_{s(\mathbf{x}_t)}) \\ & - w_s \log \mathcal{N}(\mathbf{x}_t | \eta_{s(\mathbf{x}_t)}, \Delta_{s(\mathbf{x}_t)}) \end{aligned} \tag{4}$$

where w_c, w_d and w_s are the weights. $\mathcal{N}(I(\mathbf{x}_t) | \mu^c_{s(\mathbf{x}_t)}, \Sigma^c_{s(\mathbf{x}_t)})$ describes the color distribution of segment $s(\mathbf{x}_t)$, where $\mu^c_{s(\mathbf{x}_t)}$ and $\Sigma^c_{s(\mathbf{x}_t)}$ are the mean color and covariance matrix, respectively. $\mathcal{N}(D(\mathbf{x}_t) | \mu^d_{s(\mathbf{x}_t)}, \Sigma^d_{s(\mathbf{x}_t)})$ describes the disparity distribution, which is similarly defined. $\mathcal{N}(\mathbf{x}_t | \eta_{s(\mathbf{x}_t)}, \Delta_{s(\mathbf{x}_t)})$ describes the spatial distribution of the segment $s(\mathbf{x}_t)$, where $\eta_{s(\mathbf{x}_t)}$ is the mean position coordinate, and $\Delta_{s(\mathbf{x}_t)}$ is the covariance matrix.

In order to preserve discontinuity, our spatial smoothness term is defined in an anisotropic way, encouraging the segment discontinuity to be coincident with the probabilistic boundary, color contrast and depth discontinuity. It is defined as

$$\begin{aligned} E_s(s(\mathbf{x}_t), s(\mathbf{y}_t)) = & [s(\mathbf{x}_t) \neq s(\mathbf{y}_t)] \cdot (\lambda_b \frac{\varepsilon_b}{p_b(\mathbf{x}_t, \mathbf{y}_t) + \varepsilon_b} \\ & + \lambda_c \frac{\varepsilon_c}{\|I(\mathbf{x}_t) - I(\mathbf{y}_t)\| + \varepsilon_c} \\ & + \lambda_d \frac{\varepsilon_d}{\|D(\mathbf{x}_t) - D(\mathbf{y}_t)\| + \varepsilon_d}) \end{aligned} \tag{5}$$

where λ_b, λ_c and λ_d are the smoothness weights. ε_b, ε_c, and ε_d control the contrast sensitivity.

Since it is a labeling problem, we can use belief propagation algorithm to solve (3) for spatial segmentation. We only need to solve the segment labeling of the pixels in Φ_t, and the segment labels of other pixels are all fixed. In our experiments, the 2D segment number for each frame is around 300 $\sim$ 2000. So it

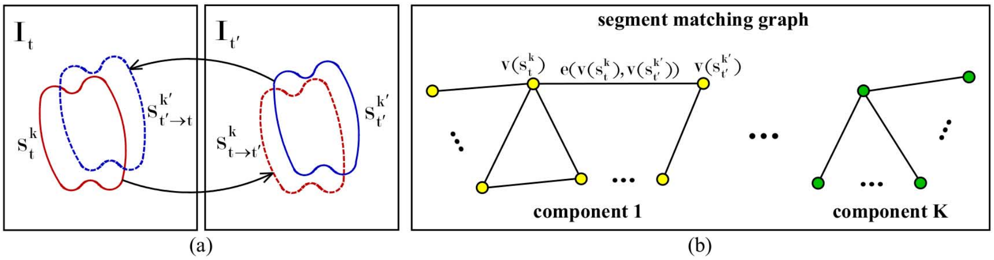


Fig. 4. Segment matching and linking. (a) Segments s_t^k and $s_{t'}^{k'}$ are projected to frame t' and t, respectively, for segment matching. (b) The connected segment components. Each component represents a volume segment.

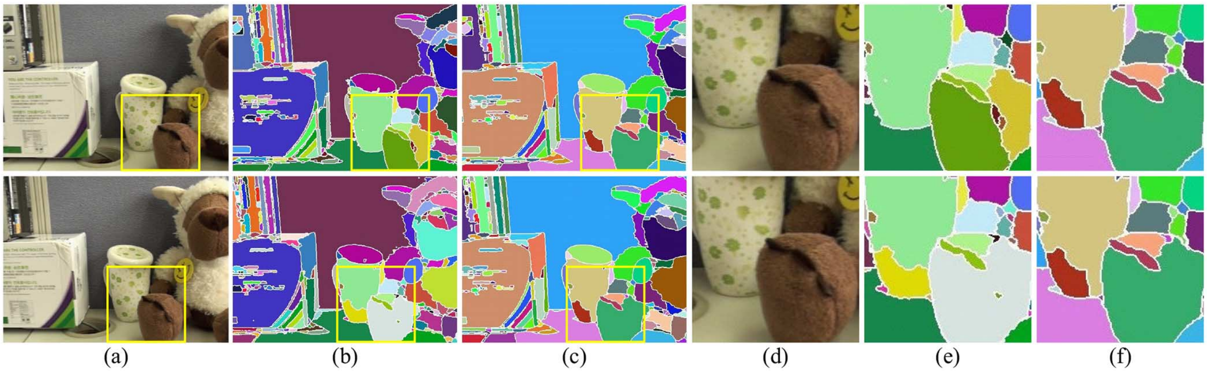


Fig. 5. Spatio-temporal segmentation results of "Desktop" sequence. (a) Two selected original images. (b) The initialized volume segments. The pixels in the same volume segment are represented with the same color. (c) The final volume segments after iterative optimization, which become more consistent and better preserve object boundaries. (d)–(f) The magnified regions of (a)–(c), highlighted with yellow rectangles.

will be very time-consuming and requires a very large memory space if we use a standard belief propagation algorithm like [47] to solve (3). In order to speed up and break through the limitation of memory space, we perform label pruning. In fact, only a small number of labels need to be considered for each pixel, since the cost of most labels are very large. Therefore, for each pixel, we only consider a few closest segments (70 segments in our experiments) with similar colors and depths. Since the time complexity of BP is linear to the number of labels [47], this label pruning strategy can well address the limitation of memory space and dramatically accelerate BP optimization, without affecting the segmentation results. The spatial segmentation results are shown in Figs. 2(d) and 3(d). The segmentation results in different images are rather consistent, which provide a good starting point for the following spatio-temporal segmentation.

V. Spatio-Temporal Segmentation

Due to the lack of explicit temporal coherence constraint, the spatial segmentation results may contain inconsistent segments. In addition, the segments in different images are not matched. In the following stage, we will perform spatio-temporal segmentation to achieve a set of pixel volumes. First, we need to match the segments in different images and link them to initialize volume segments.

A. Initializing Spatio-Temporal Volumes

Without loss of generality, we consider two 2D segments s_t^k in frame t and $s_{t'}^{k'}$ in frame t'. With the depths, we can project s_t^k from frame t to t', and $s_{t'}^{k'}$ from frame t' to t, respectively. The projection mask of s_t^k from frame t to t' is denoted as $s_{t\to t'}^k$, and the projection mask of $s_{t'}^{k'}$ from frame t' to t is denoted as $s_{t'\to t}^{k'}$. An illustration is shown in Fig. 4. We can use their overlapping rate to define the matching confidence. If $\min(|s_{t\to t'}^k \cap s_{t'}^{k'}|/|s_{t'}^{k'}|, |s_{t'\to t}^{k'} \cap s_t^k|/|s_t^k|) > \delta_v$, where δ_v is a threshold, we think s_t^k and $s_{t'}^{k'}$ are matched.

Each 2D segment can be projected to other frames, to find its matched segments in other frames. With these correspondences, we can build a matching graph. It is an undirected graph $G = (\mathcal{V}, \mathcal{E})$. Each 2D segment s_t^k corresponds to a vertex $v(s_t^k) \in \mathcal{V}$, and every pair of matched segments (s_t^k and $s_{t'}^{k'}$) has an edge $e(v(s_t^k), v(s_{t'}^{k'}))$ connecting them, as illustrated in Fig. 4(b). Each connected component represents a volume segment. The initialized volume segments are denoted as $S = \{S^k | k = 1, 2, \ldots, K\}$. One example is shown in Figs. 5(b) and (d). Most segments are already quite consistent. Then we perform an iterative optimization to further improve the results. The initialized volume segments are used as candidate labels for further optimization.

B. Iterative Optimization

For a pixel $\mathbf{x}$ in frame t, its corresponding pixel $\mathbf{x}_{t'}$ in frame t' can be computed by (1). Due to segmentation error, the segment labels of pixels $\mathbf{x}$ and $\mathbf{x}_{t'}$ may be different, i.e. $S(\mathbf{x}_{t'}) \neq S(\mathbf{x}_t)$. If there is no occlusion or out-of-view, each projection should correspond to a valid segment. In our experiments, we found that most of these projected segments are the same, which indicates that our initialized volume segments are already quite good. We use $P(\mathbf{x}_t)$ to denote the set of segment candidates for pixel $\mathbf{x}_t$, which includes these projected volume segments and $S(\mathbf{x}_t)$. Then, we define the segment probability of pixel $\mathbf{x}_t$ as

$$L_h(l, \mathbf{x}_t) = \frac{1}{|P(\mathbf{x}_t)|} \sum_{t'} [S(\mathbf{x}_{t'}) = l] \tag{6}$$

where $\mathbf{x}_{t'}$ is the projected pixel in frame t' of pixel $\mathbf{x}_t$. $L_h(l, \mathbf{x}_t)$ denotes the probability of each segment label l for pixel $\mathbf{x}_t$. Obviously, $L_h(l, \mathbf{x}_t)$ will be a large value if the assigned segment label l is consistent with most of the projected segments. For each pixel $\mathbf{x}_t$, we only need to consider the segment candidates in $P(\mathbf{x}_t)$, because the probabilities of other labels are all zeros.

We define the spatio-temporal segmentation energy in a video as follows:

$$E(S) = \sum_{t=1}^{n} \sum_{\mathbf{x}_t} \left(E'_d(S(\mathbf{x}_t)) + \sum_{\mathbf{y}_t \in N(\mathbf{x}_t)} E_s(S(\mathbf{x}_t), S(\mathbf{y}_t))\right) \tag{7}$$

where $N(\mathbf{x}_t)$ denotes the set of spatial neighbors of pixel $\mathbf{x}_t$ in frame t. The energy contains two components, i.e. data term E'_d and smoothness term E_s. E_s is the same as that of (5), and only E'_d is largely modified by incorporating the temporal coherence constraint in a statistical way.

The data term E'_d contains four components

$$\begin{aligned} E'_d(S(\mathbf{x}_t), \mathbf{x}_t) = &-w'_h \log L_h(S(\mathbf{x}_t), \mathbf{x}_t) \\ &- w'_c \log L_c(S(\mathbf{x}_t), \mathbf{x}_t) \\ &- w'_d \log L_d(S(\mathbf{x}_t), \mathbf{x}_t) \\ &- w'_s \log L_s(S(\mathbf{x}_t), \mathbf{x}_t) \end{aligned} \tag{8}$$

where w'_h, w'_c, w'_d and w'_s are the cost weights. $L_c(S(\mathbf{x}_t), \mathbf{x}_t)$ describes the Gaussian distribution of color, and is simply defined as

$$L_c(S(\mathbf{x}_t), \mathbf{x}_t) = \mathcal{N}(I(\mathbf{x}_t) | \mu^c_{S(\mathbf{x}_t)}, \Sigma^c_{S(\mathbf{x}_t)}) \tag{9}$$

where $\mu^c_{S(\mathbf{x}_t)}$ and $\Sigma^c_{S(\mathbf{x}_t)}$ are the mean color and covariance matrix of volume segment $S(\mathbf{x}_t)$, respectively. $L_d(S(\mathbf{x}_t), \mathbf{x}_t)$ describes the Gaussian distribution of disparity, and is similarly defined as

$$L_d(S(\mathbf{x}_t), \mathbf{x}_t) = \mathcal{N}(D(\mathbf{x}_t) | \mu^d_{S(\mathbf{x}_t)}, \Sigma^d_{S(\mathbf{x}_t)}) \tag{10}$$

where $\mu^d_{S(\mathbf{x}_t)}$ and $\Sigma^d_{S(\mathbf{x}_t)}$ are the mean disparity and covariance matrix of volume segment $S(\mathbf{x}_t)$, respectively.

$L_s(S(\mathbf{x}_t), \mathbf{x}_t)$ describes the shape distribution by mixture of Gaussians, which is defined as follows:

$$L_s(S(\mathbf{x}_t), \mathbf{x}_t) = \sum_{t' \in f(S(\mathbf{x}_t))} \mathcal{N}(\mathbf{x}_{t'} | \eta_{s(\mathbf{x}_{t'})}, \Delta_{s(\mathbf{x}_{t'})}) / |f(S(\mathbf{x}_t))| \tag{11}$$

where $f(S(\mathbf{x}_t))$ denotes the frames spanned by $S(\mathbf{x}_t)$, and $\mathbf{x}_{t'}$ is the corresponding pixel in frame t' for pixel $\mathbf{x}_t$. $s(\mathbf{x}_{t'})$ is the subset of $S(\mathbf{x}_t)$ in frame t'. $\eta_{s(\mathbf{x}_{t'})}$ and $\Delta_{s(\mathbf{x}_{t'})}$ are the mean coordinate and covariance matrix of $s(\mathbf{x}_{t'})$, respectively.

With the above energy definition, we iteratively refine the segmentation results using belief propagation. Each pass starts from frame 1. While solving the segmentation for frame t, the segment labels of other frames are fixed. After solving the segmentation of frame t, the related volume segments are immediately updated. One pass completes when the segmentation of frame n is optimized. In our experiments, three passes are sufficient to produce spatially and temporally coherent volume segments. One example is shown in Fig. 5. Compared to the initialized volume segments [Fig. 6(b)], the refined volume segments [Fig. 6(c)] become more consistent and better preserve object boundaries.

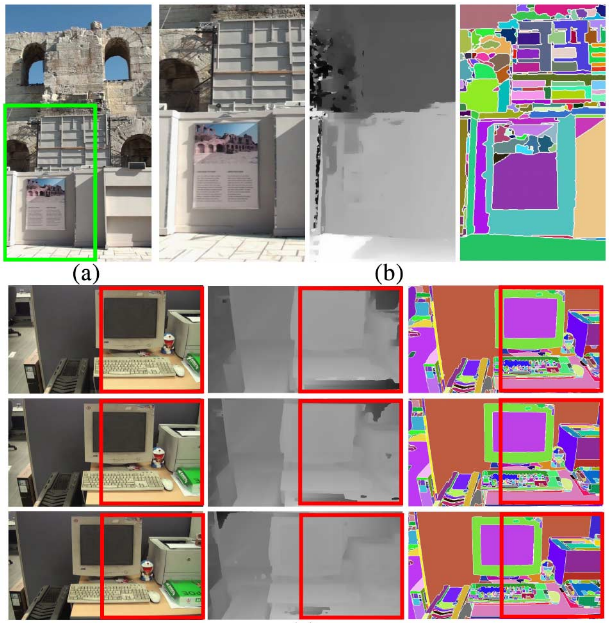


Fig. 6. Segmentation result with imperfect depth data. (a) One selected frame from "Dionysus" sequence. (b) The magnified region of (a), the estimated depths, and the segmentation result. (c) Three selected frames from "Lab" sequence, the inconsistent depth maps, and the consistent segmentation results.

VI. Experimental Results

We experimented with several challenging examples where the sequences are taken by a moving camera. The tested sequences generally contain 100 $\sim$ 200 frames. For the sequence with resolution 540×960, computing probabilistic boundary requires 26 seconds per frame on a desktop PC with Intel 4-Core 2.83 GHz CPU. The spatial segmentation requires 2 minutes per frame, and each pass of spatio-temporal optimization requires 1.5 minutes per frame. The performance of our method is acceptable for many video applications, and allows further acceleration using GPU.

The configuration of the parameters in our system is easy. Most parameters are fixed in our experiments. Specifically, $\delta_v = 0.8$, $\lambda_b = 1.33$, $\varepsilon_b = 0.6$, $\lambda_c = 0.16$, $\varepsilon_c = 0.1$, $\lambda_d = 0.16$, $\varepsilon_d = 0.1(D_{\max} - D_{\min})$. Here, $[D_{\min}, D_{\max}]$ is the disparity range of the scene. For spatial segmentation, we set $w_c = 0.54$, $w_d = 0.1$, $w_s = 0.36$. For spatio-temporal segmentation, we set $w'_h = 0.9$, $w'_c = 0.054$, $w'_d = 0.01$, $w'_s = 0.036$. Since mean shift allows the control of segmentation granularity, we can obtain different numbers of volume segments by adjusting the parameters of mean shift in the initialization stage.

A. Segmentation Results of Ordinary and Low-Frame-Rate Video Sequences

We have experimented many ordinary and low-frame-rate sequences. Please refer to our supplementary video[1] for the complete frames and results. Our segmentation method has moderate tolerance to depth estimation error, as shown in Fig. 6. Although the estimated depth maps contain noticeable artifacts as

[1] The supplementary video can be found at: http://www.cad.zju.edu.cn/home/gfzhang/projects/coseg/TMM-video.rar.

shown in Fig. 6(c), the segmentation results still preserve accurate object boundaries and are quite temporally consistent in the whole sequence. The reason is that our method mainly uses the depth information to connect the correspondences among multiple frames and collect the statistics information (such as probabilistic boundaries and the segment probability) for spatio-temporal segmentation, which is more robust than directly using depth information as an additional color channel.

Fig. 7 gives a comparison of our method with Grundmann's efficient hierarchical graph-based approach [30],[2] Xu's streaming hierarchical approach [31] and Chang's Temporal Superpixels [22]. Generally, our method can achieve more temporally consistent segments, as shown in the magnified regions in Fig. 7(e). Besides, complex occlusions are better handled by our method, such as the tree trunk illustrated in Fig. 7(f). Please refer to our supplementary video for the complete comparison result.

Though our method is developed to solve the video segmentation problem, it can also handle low-frame-rate sequences that contain a relatively small number of frames with moderately wide baselines between consecutive frames. Although the "Cones" dataset from [48] contains only 9 images, our segmentation results still preserve fine structures and faithfully maintain the coherence in different images. Please refer to our supplementary video for the segmentation results.

B. *Quantitative Evaluation of Segmentation*

Our supplementary video already allows a perceptual judgment of the spatio-temporal segmentation results. To further demonstrate the effectiveness of the proposed method, we also use the metrics similar to [49] (i.e., intra-object homogeneity, depth uniformity and temporal stability) to objectively evaluate the quality of our segmentation results. We use the texture variance employed in [49] to measure intra-object homogeneity, and use the projection overlapping rate to measure the temporal stability. All the metrics are normalized to [0, 1], and higher values indicate better results.

We first give the definitions of texture variance and depth uniformity metrics. Both kinds of metrics are normalized to the [0, 1] range, using the following formula employed in [49]:

$$v_n = \left(\frac{1}{1 + v_m/v_t} - 0.5 \right) \cdot 2 \tag{12}$$

where v_n denotes the normalized metric, v_m is the original metric value, and v_t is a truncation value determined empirically or by the nature of the metric. In our experiments, the truncation values are set to 256 and $0.2(D_{\max} - D_{\min})$ for the texture variance and depth uniformity metrics, respectively.

For video segmentation, v_m is computed by the weighted average metric of all the individual segments

$$v_m = \sqrt{\sum_{t=1}^{n} \sum_{k=1}^{K_t} w(s_t^k) M(s_t^k)} \tag{13}$$

where $w(s_t^k)$ is the weight defined as: $w(s_t^k) = |s_t^k|/V$, where V denotes the number of pixels in the 3D video volume, and $M(s_t^k)$ is the metric value for segment s_t^k. Texture variance metric $M_t(s_t^k)$ measures the color variance in s_t^k, as defined in [49], while depth uniformity metric $M_d(s_t^k)$ collects the statistics of depth boundaries (i.e., depth maps convolved with Sobel operator) contained inside the segment s_t^k, which is defined as

$$M_d(s_t^k) = \frac{1}{|\Omega(s_t^k)|} \sum_{\mathbf{x}_t \in \Omega(s_t^k)} Sobel(D(\mathbf{x}_t))^2 \tag{14}$$

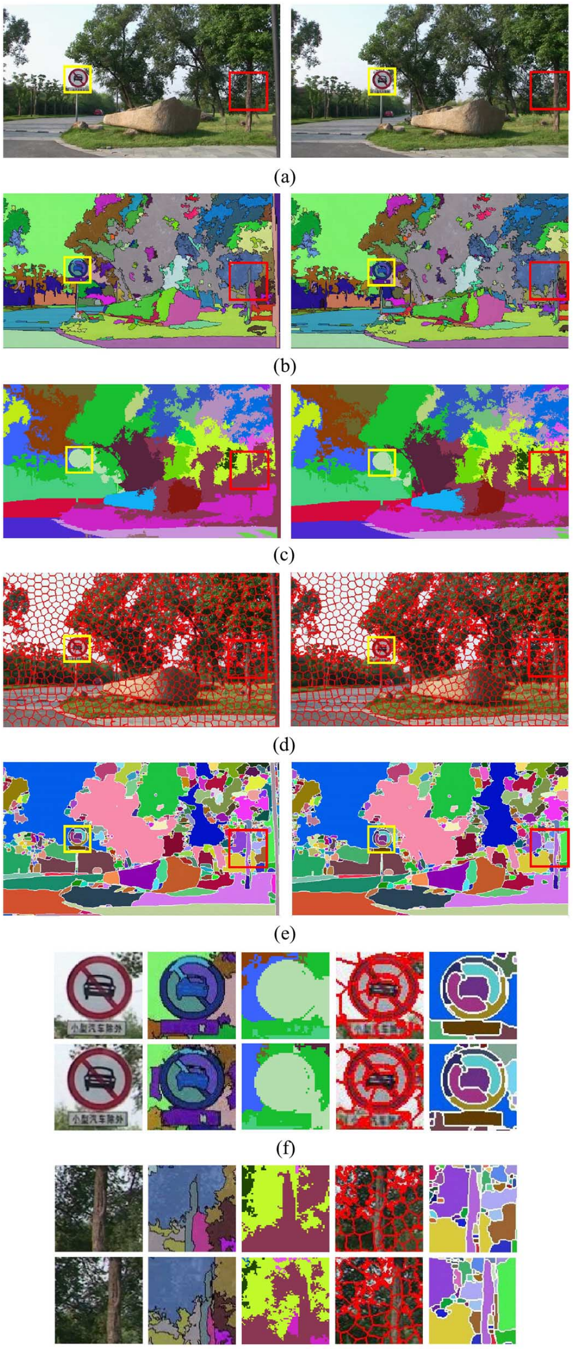


Fig. 7. Comparison to other spatio-temporal segmentation approaches [22], [30], [31]. (a) Two selected frames. (b) The spatio-temporal segmentation by Grundmann's method [30]. (c) The spatio-temporal segmentation by Xu's method [31]. (d) The spatio-temporal segmentation by Chang's method [22]. (e) Our segmentation results. (f) The magnified the regions of the yellow rectangles in (a)–(e), showing the better temporal coherence of our method. (g) The magnified red rectangles in (a)–(e), showing our better occlusion handling.

[2] We used the authors' segmentation web-service at: http://neumann.cc.gt.atl.ga.us/segmentation/.

TABLE I
QUANTITATIVE EVALUATIONS OF ALL THE TESTED SEQUENCES IN THE PAPER

Sequences		Texture Variance	Depth Uniformity	Overlapping Rate
Desktop	MS	0.88	0.91	69.47%
	SS	0.89	0.91	78.76%
	STS	0.89	0.93	87.58%
Building	MS	0.93	0.23	69.59%
	SS	0.90	0.64	81.74%
	STS	0.90	0.82	92.14%
Campus	MS	0.88	0.64	65.62%
	SS	0.89	0.78	79.23%
	STS	0.88	0.83	91.69%
Road	MS	0.85	0.84	67.70%
	SS	0.86	0.91	79.35%
	STS	0.84	0.92	92.43%
Angkor Wat	MS	0.88	0.78	64.64%
	SS	0.89	0.89	78.24%
	STS	0.88	0.90	91.66%
Great Wall	MS	0.90	0.80	58.69%
	SS	0.91	0.88	71.04%
	STS	0.90	0.89	88.08%
Garden	MS	0.82	0.76	58.96%
	SS	0.82	0.80	61.50%
	STS	0.80	0.81	88.73%
Cones	MS	0.90	0.89	72.95%
	SS	0.91	0.90	88.52%
	STS	0.91	0.90	94.61%
Dionysus	MS	0.88	0.79	60.93%
	SS	0.88	0.80	76.63%
	STS	0.87	0.82	86.74%
Carving	MS	0.91	0.89	61.34%
	SS	0.93	0.95	83.10%
	STS	0.93	0.96	92.20%
Lab	MS	0.90	0.90	82.96%
	SS	0.89	0.89	86.87%
	STS	0.90	0.89	93.08%

where $\Omega(s_t^k)$ denotes the interior of s_t^k excluding the boundary.

For measuring temporal stability, we can use the projection overlapping rate as introduced in Section V-A. The overall projection overlapping rate is computed as follows:

$$\sum_{t=1}^{n}\sum_{k=1}^{K_t}\frac{w(s_t^k)}{|N(t)|}\sum_{t'\in N(t)}\frac{|s_{t\to t'}^k \cap s_{t'}^{k'}|}{\max(|s_{t\to t'}^k|, |s_{t'}^{k'}|)} \tag{15}$$

where $N(t)$ denotes the neighboring frames of frame t (40 nearest neighboring frames in our experiment). The corresponding segment $s_{t'}^{k'}$ is the one that has the largest overlapping rate with $s_{t\to t'}^k$, which is determined by the following formula:

$$s_{t'}^{k'} = \arg\max_{s_{t'}^i \in s_{t'}} \frac{|s_{t\to t'}^k \cap s_{t'}^i|}{\max(|s_{t\to t'}^k|, |s_{t'}^i|)}. \tag{16}$$

Table I shows the three kinds of metrics on all the tested sequences in our paper. For each sequence, we evaluate the results of image-based mean shift segmentation (MS) [6], our spatial segmentation with probabilistic boundary (SS) and our iterative spatio-temporal segmentation (STS). As can be seen, the segmentation results by our spatial segmentation method have comparable texture variance with mean shift, and significantly improve the depth uniformity and temporal stability. After iterative optimization, the temporal stability is further significantly improved.

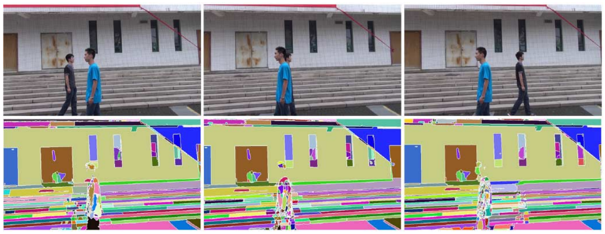


Fig. 8. Segmentation results of "Walking" sequence. Top: Three selected frames. Bottom: The extracted volume segments represented with unique color.

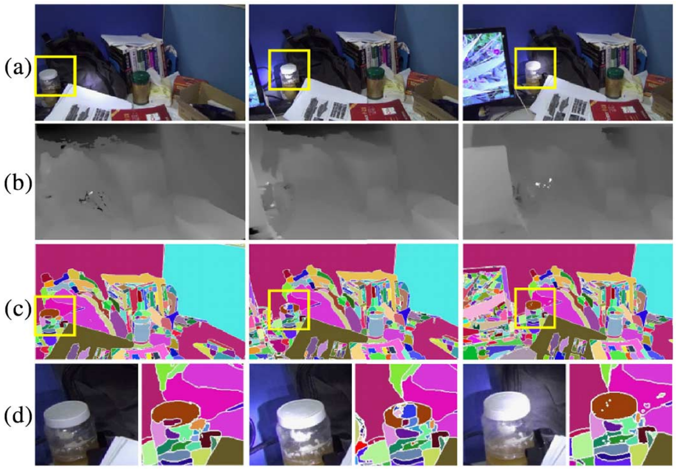


Fig. 9. Segmentation result of "Lighting Variation" sequence. (a) Three selected frames. (b) The estimated depth maps. (c) The segmentation results. (d) The magnified regions of (a) and (c).

C. Special Cases Discussion

If the sequence is captured by a static or rotating camera, the depth information cannot be recovered. Even in this case, our method still works by simply assigning constant depth for all pixels so the probabilistic boundary map computation and spatio-temporal optimization still can be performed. Our supplementary video includes a video sequence ("Rotation" sequence) captured by a rotating camera. Our method can faithfully obtain a set of spatio-temporally consistent segments for this example.

Our method is restricted to a static scene. So if there are dynamic foreground objects in the scene, their segmentation results may be inconsistent or even erroneous. One dynamic example is shown in Fig. 8. Two men walk towards each other at different depths and one occludes the other in some frames. The obtained foreground segments are indeed inconsistent, and some of them are fused into the background segments. The produced background segments are still quite consistent and not influenced much by dynamic foreground. "Hand-Waving" sequence in our supplementary video shows a case with little foreground motion (i.e., a man is waving his right hand). Our method can extract spatio-temporal segments for the static background and the man's body, while the produced segments in the moving hand are inconsistent. If there is strong lighting variation, both the appearance and estimated depth information of the influenced areas will be changed, so that the segmentation results of these areas may be inconsistent in different frames. Fig. 9 shows an example where two flashlights irradiate the desk in a back and forth way. Due to the significant appearance change, the produced segments of the bottle are inconsistent in different frames.

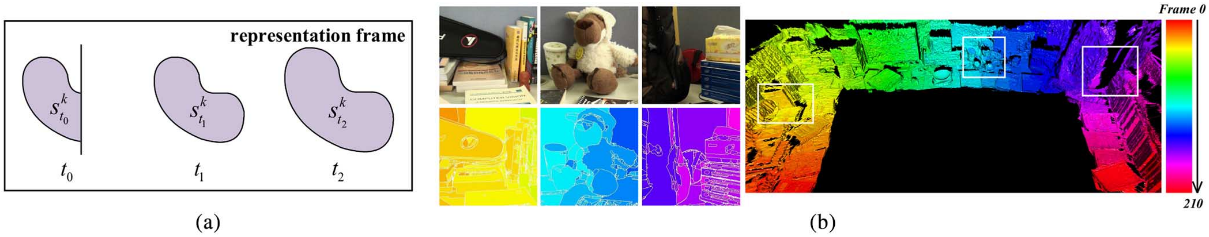


Fig. 10. (a) Representation frame selection. Volume segment S^k completely appears in frames t_1 and t_2 with full coverage rate, and has the maximal segment area in t_2. Therefore t_2 is chosen as the representation frame. (b) The representation frame map of "Desktop" sequence, visualized in 2D and 3D ways, respectively. The number of representation frame is coded with unique color.

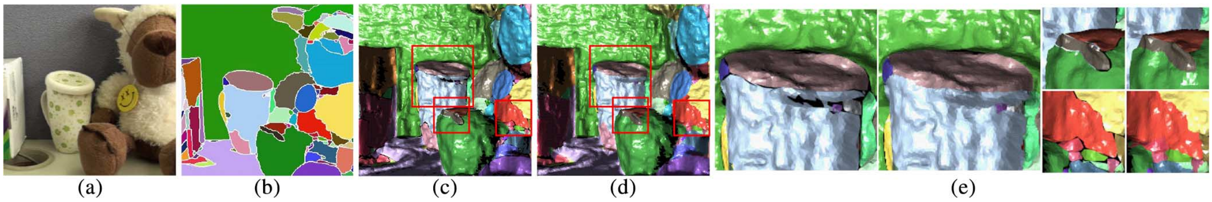


Fig. 11. Surface reconstruction and merging of volume segments. (a-b) One selected frame from the input sequence and the spatio-temporal segmentation. (c) The reconstructed 3D model without boundary merging. (d) The obtained 3D model by merging common boundaries of the neighboring segments. (e) The magnified red rectangles in (c) and (d). Seams and holes are significantly reduced after segment merging.

VII. Applications

A. 3D Geometry Reconstruction

With the spatio-temporally consistent segmentation results, the process of fusing multiple depth maps and obtaining complete 3D geometry model can be greatly facilitated.

Since the neighboring depth maps contain large overlapping content, it will result in large data redundancy if we directly fuse all depth maps to obtain the geometry model. For each volume segment S^k, its corresponding 2D regions in different frames should be consistent to each other. In most cases, we can select a frame where the corresponding 2D region can represent the whole volume segment (i.e., all pixels in the volume segment can find their correspondences in this frame). We define this frame as the representative frame for S^k. We can simply triangulate the corresponding 2D pixels (with depth information) in the representative frame to construct a mesh surface.

There are two criteria for representation frame selection. First, this frame should have large segment area so that the reconstructed mesh surface can faithfully preserve the geometry details. Second, in order to guarantee the completeness of the reconstructed geometry model, the volume segment should be fully visible in this frame, without any occlusion or out-of-view. For volume segment S^k and view t, we compute the coverage percentage $p^t_c(S^k)$ by the average projection overlapping rate (as introduced in Section V-A) of s^k_t with other frames, which is defined as

$$p^t_c(S^k) = \sum_{t' \in f(S^k)} |s^k_{t \to t'} \cap s^k_{t'}| / \sum_{t' \in f(S^k)} |s^k_{t'}| \qquad (17)$$

where $f(S^k)$ is defined the same as in (11). If $p^t_c(S^k)$ is low, there are some occlusions or out-of-view for 2D segment s^k_t in t, so that frame t is not in the candidates of representation frames of S^k.

We first select a set of candidate frames which satisfy $p^t_c(S^k) > 0.98$, as illustrated in Fig. 10(a). Then the frame with maximal number of pixels is selected as the representation frame. Fig. 10(b) shows an example of representation frame selection. Each segment is encoded with a color, indexing the frame number of the selected representation frame.

Due to serious occlusion or out-of-view, there are some segments which do not satisfy the hypothesis that each volume segment can always find at least one frame where the segment is fully visible. For these segments, we need to split each of them to a number of smaller segments which can satisfy the hypothesis. To split volume segment S^k, we first assign all the 2D pixels of S^k as unlabeled. The set of all unlabeled pixels is denoted as U^k, and the set of unlabeled pixels in frame t is denoted by u^k_t. Then we iteratively split the segment and create new segments. In each iteration, we find a frame t_i which has largest projection overlapping rate $p^{t_i}_c(U^k)$, defined by

$$p^t_c(U^k) = \sum_{t' \in f(S^k)} |u^k_{t \to t'} \cap u^k_{t'}| / \sum_{t' \in f(S^k)} |u^k_{t'}|. \qquad (18)$$

Then we create a new segment $S^{k'}$ which includes u^k_t and its projections in neighboring frames. This process is repeated until U^k only contains a small number of pixels ($|U^k| < 0.01|S^k|$ in our experiment), which are caused by segment noises around the boundaries. These unlabeled pixels are finally fused to the neighboring segments using the method introduced in Section IV-B.

After representation frame selection, we triangulate the pixels of each segment in its representation frame. Then 3D surface of each segment can be constructed by projecting the pixels to 3D space with depth information. In order to avoid unexpected gaps along the boundaries between neighboring 3D surfaces, as shown in Figs. 11(c) and (e), we need to connect the neighboring 3D surfaces by merging their common 3D boundaries.

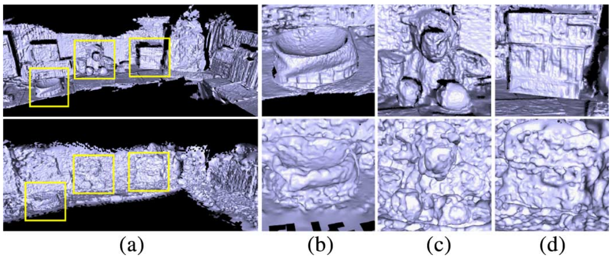


Fig. 12. Comparison to PMVS2 [4] on "Desktop" sequence. (a) Our reconstruction result (top) and the result by PMVS2 (bottom). (b)–(d) The magnified surface of the regions in (a), highlighted with yellow rectangles.

Both segmentation and depth information can be utilized to verify whether two volume segments share a common boundary. If two neighboring volume segments have common 2D boundaries in their respective representation frames, and the common 2D boundaries should be depth continuous, we consider these two volume segments as neighbors. For each volume segment S^k with its representation frame t_o, we check all the 2D neighbor segments $N(s_{t_o}^k)$ in frame t_o. For a 2D segment $s_{t_o}^{k'} \in N(s_{t_o}^k)$, if most of the pixels (90% pixels in our experiments) along the common 2D boundary are depth continuous, we think $S^{k'}$ and S^k are neighbors, and share a common 3D boundary. Similar to [50], the depth continuity of two neighboring pixels $\mathbf{x}_o$ and $\mathbf{x}'_o$ can be verified by the following criteria:

$$2|z(\mathbf{x}_o) - z(\mathbf{x}'_o)|/(z(\mathbf{x}_o) + z(\mathbf{x}'_o)) < \lambda_z \tag{19}$$

where λ_z is a threshold and set to 0.03 in our experiments. After merging neighboring volume segments, we apply Laplacian surface smoothing [51] method to refine the unsmooth boundaries caused by depth inconsistency in different frames, and detect and fill small isolated holes with fewer than 200 edges to complete the surface mesh. As shown in Fig. 11(d) and (e), most unwanted seams are removed by merging common boundaries. Fig. 12 gives a comparison between our reconstruction method and PMVS2 [4] with Poisson surface reconstruction [52]. As can be seen in the magnified regions in Fig. 12(b)–(d), our reconstructed model can preserve better geometry details.

Fig. 13 shows the reconstructed scene geometry of the "Dionysus" example. Our method can naturally handle the 3D reconstruction of large-scale scenes, because we can always perform spatio-temporal segmentation first, and separate the whole scene into a set of relatively small volume segments. Since the 3D geometry of each segment can be reconstructed independently, and then connected to construct a complete 3D geometry, our method can handle large-scale 3D reconstruction with limited memory. In contrast, large-scale 3D reconstruction is difficult for voxel-based methods [52], [53].

B. Video Editing

Given the 3D geometry model incorporating segmentation labeling, we can easily mark out the objects of interest for cloning with a few user interactions (e.g., draw a rectangle). The 3D segmentation labeling greatly facilitates the object selection and cloning, as demonstrated in Fig. 14 and our supplementary video. The cloned objects can be added to the designated position in the original scene model with simple user interactions. We can also render them to the original video for novel video synthesis, as shown in Figs. 14(d)–(e). The occlusions can be naturally handled with the 3D information. Please refer to our supplementary video that gives a better presentation of the edited results on both "Desktop" and "Dionysus" examples.

C. Video Stylization

Video stylization and abstraction are useful in many application areas, such as broadcast and communications, video games, and many other entertainments [54]. The temporal coherence of stylization effects is very important in such applications. With our segmentation results, spatio-temporally consistent stylized effects can be faithfully created. An example is shown in Fig. 15, using a method similar to [20], [54]. Each region is represented by its mean color, with DoG-edges overlayed. We first presmooth each frame by bilateral filtering [55], and then use the method proposed in [54] to detect DoG edges. The spatio-temporal consistency of our segmentation can guarantee the high coherence of non-photorealistic stylization effects [Fig. 15(b)]. In contrast, if we use mean-shift [Fig. 15(c)] to segment each frame independently, the obtained stylization effects is quite flickering as shown in our supplementary video.

D. Consistent Semantic Segmentation

Our spatio-temporal segmentation results can be directly applied to semantic segmentation. Since our video segmentations are consistent among temporal frames, corresponding segments across different frames can be jointly segmented into a same semantic label, which makes semantic segmentation easier and produces more consistent semantic labeling.

To perform semantic segmentation on "Campus" sequence, we use another sequence capturing a similar scene for feature training, as in Fig. 16(a). We use SLIC [56] to segment each frame of the training sequence into superpixels [Fig. 16(b)], and extract five kinds of features for each superpixel. The first four features are extracted from the dense depth values of each superpixel as in [57], which are surface normal, surface local planarity, height above ground and distance to camera path respectively. The fifth feature is the RGB color histogram of the superpixel. The five features are cascaded as a high-dimensional descriptor for each superpixel. Besides, the superpixels of each frame of the training sequence are manually classified into several predefined semantic objects, as shown in Fig. 16(c). All the descriptors and semantic object segments are used to train a semantic classifier using randomized decision forest [58]. Finally, with the spatio-temporal segmentation result of the "Campus" sequence, we use the trained randomized decision forest classifier to decide the best semantic label for each volume segment in "Campus" sequence.

The semantic segmentation results are shown in Fig. 16(e), which are both accurate and consistent among temporal frames. Another semantic segmentation example of "Building" sequence is included in the supplementary video.

VIII. Conclusions and Discussion

In this paper, we have proposed a novel video segmentation method, which can extract a set of spatio-temporal volume segments from a depth-inferred video. Most previous approaches

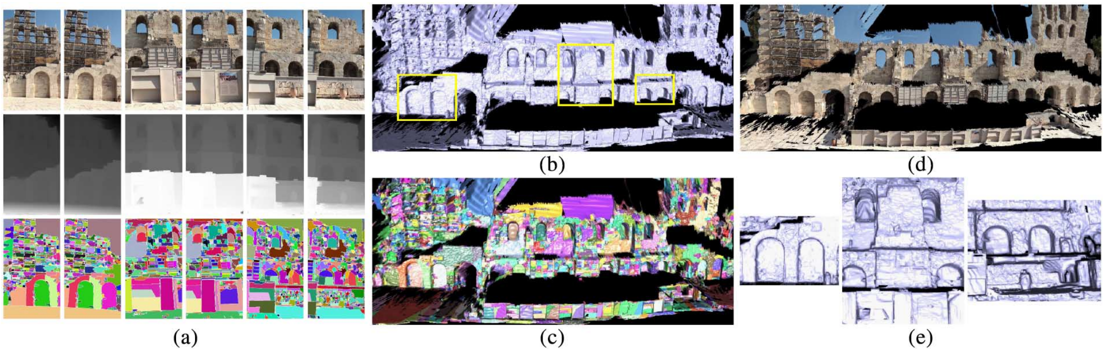


Fig. 13. Spatio-temporal segmentation and 3D reconstruction of the "Dionysus" sequence, which captures the "Theatre of Dionysus" in the Acropolis of Athens. (a) Selected original frames, the recovered depth maps and the spatio-temporal consistent segmentation results. The extracted volume segments are represented with unique color. (b) The reconstructed 3D scene by fusing depth maps with the consistent segmentation. (c) The reconstructed volume segments represented with unique color. (d) The texture-mapped scene model. (e) The magnified reconstructed surface mesh of the regions in (b), highlighted with yellow rectangles.

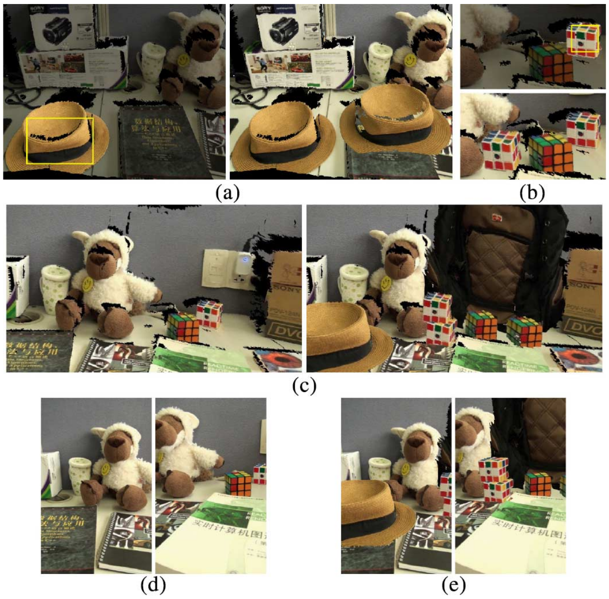


Fig. 14. Scene/video editing of "Desktop" sequence. (a)–(c) Object selection and cloning. The 3D segmentation information significantly facilitates the object selection. The user only needs to draw a rough rectangle to quickly select the hat or magic cube. (d) Two selected video frames. (e) The composite frames after editing.

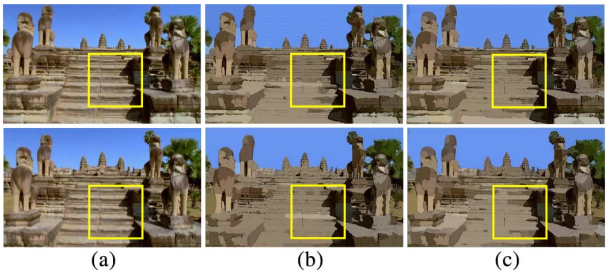


Fig. 15. Non-photorealistic video stylization effects of "Angkor Wat" sequence. (a) Two selected original images. (b) The stylization effects of mean-shift, which are flickering among different frames, as highlighted in yellow rectangles. (c) The highly consistent stylization of our segmentation without flickering.

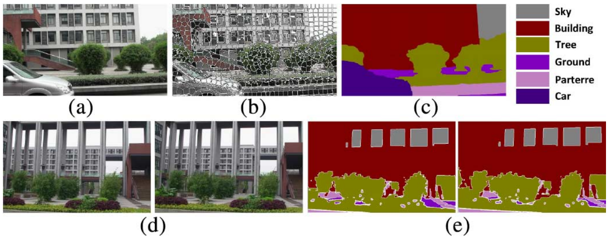


Fig. 16. Consistent semantic segmentation of "Campus" sequence. (a) One frame of the training sequence. (b) SLIC Superpixels of (a). (c) Manually marked semantic classification of (b). (d) Two selected original images of "Campus" sequence. (e) The consistent semantic segmentation results of (d).

rely on pairwise motion estimation, which are sensitive to large displacement with occlusions. By utilizing depth information, we can connect the correspondences among multiple frames, so that the statistics information, such as probabilistic boundaries and the segment probability of each pixel, can be effectively collected. By incorporating these statistics information into the segmentation energy function, our method can robustly handle significant occlusions, so that a set of spatio-temporally consistent segments can be achieved. Using our spatio-temporal segmentation, we can reconstruct 3D geometry models for large-scale scenes containing 3D segmentation information, which is useful in many other applications such as video editing/stylization and semantic segmentation.

Our method uses a single handheld camera and multiview stereo method to recover the depth maps and collect multi-frame statistics, which is restricted to videos of a static scene. We believe our method can be improved in the future to handle dynamic scenes. For moving objects, the temporal correspondences among the different frames should be built by motion estimation or 3D scene flow tracking with a depth camera or multiple synchronized video cameras, so that the temporally coherence constraint can be reliably enforced. In addition, given an extremely small number of wide-baseline images, the collected statistics may be degraded and handling the problems of large occlusions or out-of-view will become more difficult, which may cause our method to produce unsatisfactory segmentation results. Fig. 4 in our supplementary document[3] shows a failure example. How to solve this problem remains to be our future work.

[3] The supplementary document can be found at: http://www.cad.zju.edu.cn/home/gfzhang/projects/coseg/TMM-supple.pdf.

Acknowledgment

The authors would like to thank all the reviewers for their constructive comments to improve this paper, and J. Zheng for his kind help in this work.

References

[1] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2004, ISBN: 0521540518.

[2] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *Proc. CVPR*, 2006, vol. 1, pp. 519–528 [Online]. Available: http://vision.middlebury.edu/mview/

[3] G. Zhang, J. Jia, T.-T. Wong, and H. Bao, "Consistent depth maps recovery from a video sequence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 6, pp. 974–988, Jun. 2009.

[4] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 8, pp. 1362–1376, Aug. 2010.

[5] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[6] D. Comaniciu, P. Meer, and S. Member, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.

[7] H. Mobahi, S. Rao, A. Y. Yang, S. S. Sastry, and Y. Ma, "Segmentation of natural images by texture and boundary compression," *Int. J. Comput. Vis.*, vol. 95, no. 1, pp. 86–98, 2011.

[8] E. Sharon, M. Galun, D. Sharon, R. Basri, and A. Brandt, "Hierarchy and adaptivity in segmenting visual scenes," *Nature*, vol. 442, no. 7104, pp. 719–846, Jun. 2006.

[9] R. Megret and D. Dementhon, "A survey of spatio-temporal grouping techniques," Language and Media Process., Univ. of Maryland, College Park, MD, USA, Tech. Rep. LAMP-TR-094/CS-TR-4403, 2002.

[10] M. P. Kumar, P. H. S. Torr, and A. Zisserman, "Learning layered motion segmentations of video," *Int. J. Comput. Vis.*, vol. 76, no. 3, pp. 301–319, 2008.

[11] J. Shi and J. Malik, "Motion segmentation and tracking using normalized cuts," in *Proc. ICCV*, 1998, pp. 1154–1160.

[12] C. Fowlkes, S. Belongie, and J. Malik, "Efficient spatiotemporal grouping using the nystrom method," in *Proc. CVPR*, 2001, pp. 231–238.

[13] S. Khan and M. Shah, "Object based segmentation of video using color, motion and spatial information," in *Proc. CVPR*, 2001, vol. 2, pp. 746–750.

[14] D. Cremers and S. Soatto, "Variational space-time motion segmentation," in *Proc. ICCV*, 2003, pp. 886–893.

[15] C. L. Zitnick, N. Jojic, and S. B. Kang, "Consistent segmentation for optical flow estimation," in *Proc. ICCV*, 2005, vol. 2, pp. 1308–1315.

[16] Y. Deng and B. S. Manjunath, "Unsupervised segmentation of color-texture regions in images and video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 8, pp. 800–810, Aug. 2001.

[17] W. Brendel and S. Todorovic, "Video object segmentation by tracking regions," in *Proc. ICCV*, 2009, pp. 833–840.

[18] S. Liu, G. Dong, C. H. Yan, and S. H. Ong, "Video segmentation: Propagation, validation and aggregation of a preceding graph," in *Proc. CVPR*, 2008, pp. 1–7.

[19] Y. Wang and Q. Ji, "A dynamic conditional random field model for object segmentation in image sequences," in *Proc. CVPR*, 2005, vol. 1, pp. 264–270.

[20] S. Zhang, X. Li, S. Hu, and R. R. Martin, "Online video stream abstraction and stylization," *IEEE Trans. Multimedia*, vol. 13, no. 6, pp. 1268–1294, Dec. 2011.

[21] M. V. den Bergh, G. Roig, X. Boix, S. Manen, and L. V. Gool, "Online video seeds for temporal window objectness," in *Proc. ICCV*, 2013, pp. 377–384.

[22] J. Chang and D. WeiJ. W. F. , III, "A video representation using temporal superpixels," in *Proc. CVPR*, 2013, pp. 2051–2058.

[23] A. Vázquez-Reina, S. Avidan, H. Pfister, and E. Miller, "Multiple hypothesis video segmentation from superpixel flows," in *Proc. ECCV*, 2010, pp. 268–281.

[24] F. Galasso, R. Cipolla, and B. Schiele, "Video segmentation with superpixels," in *Proc. ACCV*, 2012, pp. 760–774.

[25] A. Abramov, K. Pauwels, J. Papon, F. Worgotter, and B. Dellen, "Depth-supported real-time video segmentation with the kinect," in *Proc. IEEE Workshop Appl. Comput. Vis.*, Jan. 2012, pp. 457–464.

[26] D. Dementhon and R. Megret, "Spatio-temporal segmentation of video by hierarchical mean shift analysis," Univ. of Maryland, College Park, MD, USA, Tech. Rep. LAMP-TR-090/CAR-TR-978/CS-TR-4388/UMIACS-TR-2002-68, 2002.

[27] J. Wang, B. Thiesson, Y. Xu, and M. Cohen, "Image and video segmentation by anisotropic kernel mean shift," in *Proc. ECCV*, 2004, pp. 238–249.

[28] H. Greenspan, J. Goldberger, and A. Mayer, "Probabilistic space-time video modeling via piecewise GMM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 3, pp. 384–396, Mar. 2004.

[29] M. Reso, J. Jachalsky, B. Rosenhahn, and J. Ostermann, "Temporally consistent superpixels," in *Proc. ICCV*, 2013, pp. 385–392.

[30] M. Grundmann, V. Kwatra, M. Han, and I. Essa, "Efficient hierarchical graph based video segmentation," in *Proc. CVPR*, 2010, pp. 2141–2148.

[31] C. Xu, C. Xiong, and J. J. Corso, "Streaming hierarchical video segmentation," in *Proc. ECCV*, 2012, pp. 626–639.

[32] J. Lezama, K. Alahari, J. Sivic, and I. Laptev, "Track to the future: Spatio-temporal video segmentation with long-range motion cues," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 3369–3376.

[33] L. Quan, J. Wang, P. Tan, and L. Yuan, "Image-based modeling by joint segmentation," *Int. J. Comput. Vis.*, vol. 75, no. 1, pp. 135–150, 2007.

[34] J. Xiao, J. Wang, P. Tan, and L. Quan, "Joint affinity propagation for multiple view segmentation," in *Proc. ICCV*, 2007, pp. 1–7.

[35] D. Gallup, J.-M. Frahm, and M. Pollefeys, "Piecewise planar and non-planar stereo for urban scene reconstruction," in *Proc. CVPR*, 2010, pp. 1418–1425.

[36] P. Sand and S. J. Teller, "Video matching," *ACM Trans. Graphics*, vol. 23, no. 3, pp. 592–599, 2004.

[37] J. Wang, Y. Xu, H.-Y. Shum, and M. F. Cohen, "Video tooning," *ACM Trans. Graphics*, vol. 23, no. 3, pp. 574–583, 2004.

[38] Y. Li, J. Sun, and H.-Y. Shum, "Video object cut and paste," *ACM Trans. Graphics*, vol. 24, no. 3, pp. 595–600, 2005.

[39] Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 463–476, Mar. 2007.

[40] H.-W. Kang, Y. Matsushita, X. Tang, and X.-Q. Chen, "Space-time video montage," in *Proc. CVPR (2)*, 2006, pp. 1331–1338.

[41] J. Xiao, X. Cao, and H. Foroosh, "3D object transfer between non-overlapping videos," in *Proc. IEEE Virtual Reality Conf.*, Mar. 2006, pp. 127–134.

[42] P. Bhat, C. L. Zitnick, N. Snavely, A. Agarwala, M. Agrawala, B. Curless, M. Cohen, and S. B. Kang, "Using photographs to enhance videos of a static scene," in *Proc. Eurograph. Symp. Rendering Tech.*, 2007, pp. 327–338.

[43] G. Zhang, Z. Dong, J. Jia, L. Wan, T.-T. Wong, and H. Bao, "Refilming with depth-inferred videos," *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 5, pp. 828–840, Sep.–Oct. 2009.

[44] S. B. Kang and R. Szeliski, "Extracting view-dependent depth maps from a collection of images," *Int. J. Comput. Vis.*, vol. 58, no. 2, pp. 139–163, 2004.

[45] G. Zhang, X. Qin, W. Hua, T.-T. Wong, P.-A. Heng, and H. Bao, "Robust metric reconstruction from challenging video sequences," in *Proc. CVPR*, 2007, pp. 1–8.

[46] P. D. Smet and R. L. V. Pires, "Implementation and analysis of an optimized rainfalling watershed algorithm," *Proc. SPIE*, vol. 3974, pp. 759–766, 2000.

[47] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient belief propagation for early vision," *Int. J. Comput. Vis.*, vol. 70, no. 1, pp. 41–54, 2006.

[48] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," in *Proc. CVPR*, 2003, vol. 1, pp. 195–202.

[49] P. L. Correia and F. Pereira, "Objective evaluation of video segmentation quality," *IEEE Trans. Image Process.*, vol. 12, no. 2, pp. 186–200, Feb. 2003.

[50] R. Pajarola, M. Sainz, and Y. Meng, "Depth-mesh objects: Fast depth-image meshing and warping," School of Inf. and Comput. Sci., Univ. of California Irvine, Irvine, CA, USA, Tech. Rep. UCI-ICS-03-02, 2003.

[51] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel, "Laplacian surface editing," in *Proc. Eurograph./ACM SIGGRAPH Symp. Geom. Process.*, 2004, pp. 179–188.

[52] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proc. 4th Eurograph. Symp. Geom. Process.*, 2006, pp. 61–70.

[53] S. M. Seitz and C. R. Dyer, "Photorealistic scene reconstruction by voxel coloring," *Int. J. Comput. Vis.*, vol. 35, no. 2, pp. 151–173, 1999.

[54] W. Holger, O. S. C. , and G. Bruce, "Real-time video abstraction," *ACM Trans. Graphics*, vol. 25, no. 3, pp. 1221–1226, 2006.

[55] P. Sylvain and D. Frédo, "A fast approximation of the bilateral filter using a signal processing approach," *Int. J. Comput. Vis.*, vol. 81, no. 1, pp. 24–52, 2009.

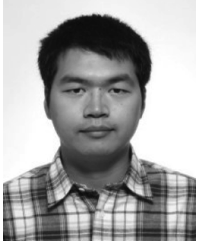
[56] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[57] C. Zhang, L. Wang, and R. Yang, "Semantic segmentation of urban scenes using dense depth maps," in *Proc. ECCV*, 2010, pp. 708–721.

[58] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, 2006.

Hanqing Jiang (S'12) received the B.S. and Ph.D. degrees in computer science and technology from Zhejiang University, Hangzhou, China, in 2006 and 2012, respectively.

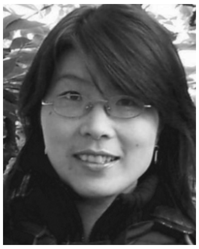
He is currently a Postdoctor at the State Key Laboratory of CAD and CG, Zhejiang University, Hangzhou, China. His research interests include video-based 3D reconstruction and segmentation.



Guofeng Zhang (S'07–M'08) received the B.S. and Ph.D. degrees in computer science and technology from Zhejiang University, Hangzhou, China, in 2003 and 2009, respectively.

He is currently an Associate Professor at State Key Laboratory of CAD and CG, Zhejiang University, Hangzhou, China. His research interests include structure-from-motion, 3D reconstruction, augmented reality, video segmentation, and editing.

Dr. Zhang was a recipient of the National Excellent Doctoral Dissertation Award and the Excellent Doctoral Dissertation Award of the China Computer Federation.



Huiyan Wang (M'09) received the M.S. degree in power engineering from Shandong University, Jinan, China and the Ph.D. degree in electrical engineering from Zhejiang University, Hangzhou, China, in 1999 and 2003, respectively, and conducted postdoctoral research in clinical medicine from the Pharmaceutical Informatics Institute, Zhejiang University, Hangzhou, China, from 2003 to 2005.

She is currently a Professor of Computer Science and Technology with the School of Computer Science and Information Engineering, Zhejiang Gongshang University, Hangzhou, China. Her current interests are biomedical signal processing and pattern recognition, and she also works on image and video processing.

Hujun Bao (A'14) received the B.S. and Ph.D. degrees in applied mathematics from Zhejiang University, Hangzhou, China, in 1987 and 1993, respectively.

He is currently a Cheung Kong Professor at State Key Laboratory of CAD and CG, Zhejiang University, Hangzhou, China. His main research interest are in computer graphics and computer vision, including geometry and vision computing, real-time rendering, and mixed reality.