Automatic Video Object Segmentation Based on Visual and Motion Saliency

Qinmu Peng, *Member, IEEE*, and Yiu-Ming Cheung, *Fellow, IEEE*

***Abstract*—We present an approach to extract the salient object automatically in videos. Given an unannotated video sequence, the proposed method first computes the visual saliency to identify object-like regions in each frame based on the proposed weighted multiple manifold ranking algorithm. We then compute motion cues to estimate the motion saliency and localization prior. Finally, adopting a new energy function, we estimate a superpixel-level object labeling across all frames, where 1) the data term depends on the visual saliency and localization prior, and 2) the smoothness term depends on the constraints in time and space. Compared to the existing counterparts, the proposed approach automatically segments the persistent foreground object meanwhile preserving the potential shape. Experiments show its promising results on the challenging benchmark videos in comparison with the existing counterparts.**

***Index Terms*—Object segmentation, visual saliency, manifold ranking, graph model.**

I. Introduction

AUTOMATIC video object segmentation (AVOS) is to separate the foreground objects from the background in a video automatically, which has a variety of potential applications, including video summarization [1], action recognition [2], image retrieval [3], and so on. Nevertheless, AVOS is a non-trivial task in computer vision and pattern recognition, although human being can easily deal with this process by the complex cognitive capabilities of human brains even though the object is presented in a complicated background or even has never been seen before [4].

To accomplish the task of AVOS, a number of approaches based on different theories and methodologies have been proposed in the literature. In general, the AVOS approaches have been developed along two lines, i.e. supervised and unsupervised approaches. The former requires the interactive operation with a user to annotate the object position in some frames. For example, Bai *et al.* [5] have adopted a set of local classifiers, each of which integrates multiple local image features such as color, texture, shape and motion. Once the initial annotations are created, they are then propagated to all other frames, and the object cutout is completed with a video matting technique. Similarly, Brian *et al.* [6] have proposed to utilize various visual cues, and each of which is automatically weighted based on the likely effectiveness. That method allows a user to segment one frame and then propagates this information to the other frames. Moreover, Tsai *et al.* [7] presented an off-line method for object segmentation and tracking by utilizing multi-label Markov random filed model, which incorporates both segmentation and motion estimation. Besides, researchers have proposed to draw a few strokes on arbitrary regions in the foreground and background for the purpose of simplifying manual annotation. For example, Wang *et al.* [8] utilized a global color model based on a user's strokes, and incorporated a local color model for backgrounds in addition to gradient values. Along this way, Bai and Sapiro [9] have proposed the weighted geodesic distances to describe the user-provided scribbles, and additional constraints are added into the distance, which could efficiently handle occlusions in a video sequence. Recently, Zhang *et al.* [10] have proposed a semantic object segmentation framework using a weakly supervised approach. The success of these works have been reported in their application domains, but it might be difficult for users to manually annotate a large amount of video data from the practical perspective.

By contrast, the latter unsupervised approaches do not require the process of manual labeling and can automatically extract the objects from the background. Along this line, classic background substraction methods have been widely used for extraction of moving objects in a video, which model the appearance of the background at each pixel and regard pixels that change rapidly as the foreground [11], [12]. Those methods usually assume that a video is captured by a static camera, in which the background changes slowly so that the model could correctly update the foreground appearance. However, if the background is changeable, modeling the background appearance then becomes a tricky

Manuscript received December 29, 2017; revised April 15, 2019; accepted May 6, 2019. Date of publication May 23, 2019; date of current version November 19, 2019. This work was supported in part by the National Natural Science Foundation of China under Grants 61672444 and 61272366, in part by the Faculty Research Grant of Hong Kong Baptist University (HKBU) under Project FRG2/17-18/082, in part by the KTO Grant of HKBU under Project MPCF-004-2017/18, and in part by the SZSTI under Grants: JCYJ20160531194006833 and JCYJ20180305180637611. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Xilin Chen. *(Corresponding author: Yiu-Ming Cheung.)*

Q. Peng is with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China, Department of Computer Science, Hong Kong Baptist University, Hong Kong, China, and also with the Shenzhen Research Institute of Huazhong University of Science and Technology, Shenzhen 518055, China (e-mail: pengqinmu@hust.edu.cn).

Y.-M. Cheung is with the Department of Computer Science, Hong Kong Baptist University, Kowloon, Hong Kong, and also with the HKBU Institute of Research and Continuing Education, Shenzhen 518057, China (e-mail: ymc@comp.hkbu.edu.hk).

This paper has supplementary downloadable materials available at http://ieeexplore.ieee.org, provided by the author. The material includes a demonstration video to show the spatial-temporal consistency between adjacent frames. This material is 982 KB 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.2019.2918730

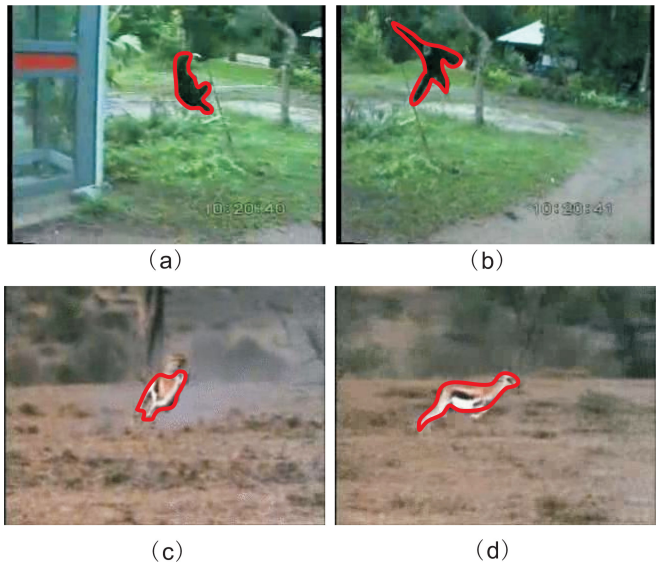


Fig. 1. Several snapshots in different video sequences. Sub-figure (a)-(b) illustrate non-rigid deformation; Sub-figure (c)-(d) demonstrate the motion blur and low contrast between the foreground and background.

issue. For such cases, a typical approach is to learn a dynamic background model, and thus the foreground objects are extracted as outliers. For example, Zhong and Sclaroff [13] utilized a robust Kalman filter to update the dynamic autoregressive moving average model iteratively, and to determine a mask image for the foreground object. Recently, other methods have been explored as well to depict the foreground object in video data. For example, Papazoglou *et al.* [14] have utilized the motion boundary and inside-outside maps based on the optical flow to segment objects in videos. Furthermore, a few methods (e.g. [15]–[17]) incorporate image saliency for video object segmentation, and several works segment moving objects by spatio-temporal segment proposals [18]–[21]. In addition, Ramakanth and Babu [22] have utilized the formulation of seams for temporal label propagation to segment the object in the video sequences. Chen *et al.* [23] proposed a moving object segmentation method in video by utilizing improved point trajectories. Recently, Tsai *et al.* [24] have proposed a joint optimization of segmentation and optical flow scheme which demonstrates its effectiveness in segmenting objects in videos. Furthermore, Wang *et al.* [25] have presented a pyramid histogram based confidence map and also combined geodesic distance based dynamic models for the video object segmentation.

Nevertheless, the general problem of the video object segmentation is still a very challenging task, especially for the unconstrained videos, in which the background may have complex transformation without a single scene, and the illumination may change. Moreover, the moving objects undergo non-rigid deformation, and suffer from motion blur or demonstrate the similar appearance to the parts of the background. Fig. 1 shows several snapshots for such examples. Under the circumstances, most of the existing methods cannot work well on such unconstrained video (e.g. [13], [26], [27]). Our goal is to extract the foreground object automatically from such video sequences without any user annotation. With the development of the human visual attention models (e.g. [28], [29]), saliency-based object detection (e.g. [30]–[38]) is one of the most promising approaches to AVOS because it is able to obtain desirable prior information for inferring the region of the foreground object in the videos. Incorporating the image or video saliency to estimate the candidate foreground object is expected to be more effective. However, estimation of the target object in the video and providing consistent and reliable priors for higher level object segmentation task is still a challenging problem. As far as we know, few works only, e.g. [39] [40], have been specifically designed for video saliency thus far. These methods usually adopt a combination of the existing image saliency models with motion cues, but the performance of these methods is insufficient to guide the accurate object segmentation in the video.

In this paper, we will develop a salient object detection approach featuring higher-detection performance and less demanding assumption. The method computes the visual saliency in each frame using a weighted multiple manifold ranking algorithm. It then computes motion cues to estimate the motion saliency and localization prior. By adopting a new energy function, the data term depends on the visual saliency and localization priors, and the smoothness term depends on the constraint in time and space. Compared to the existing counterparts, the proposed approach automatically segments the persistent foreground object meanwhile preserving the potential shape. We apply this method to challenging benchmark videos, and show competent or even better performance than the existing counterparts. In summary, the main contributions of this paper are two-fold:

1) The proposed method is a weighted multiple manifold ranking algorithm for the saliency detection, which gives higher saliency detection performance.
2) We adopt a new energy function to estimate a superpixel-level object labeling across all frames, which preserves clear appearance and shape for the salient objects in the images.

The remainder of this paper is organized as follows: Section II provides the overview of manifold ranking model, and the graph-based segmentation. The details of the proposed video object segmentation method is presented in Section III. In Section IV, we will evaluate the performance of the proposed approach in comparison with the existing counterparts. Finally, we draw a conclusion in Section V.

II. Overview of Manifold Ranking and Graph-Based Segmentation

This section will make an overview of the manifold ranking model and the graph-based segmentation.

Manifold Ranking: Manifold ranking is to measure relevance between the query and the remaining data, which is typically represented by a weighted graph. The queries are assigned a positive value and the remaining nodes are ranked with respect to the queries. He *et al.* [41] were the first attempt to apply manifold ranking to the image retrieval and obtained promising result. In manifold ranking, an image is mapped into a

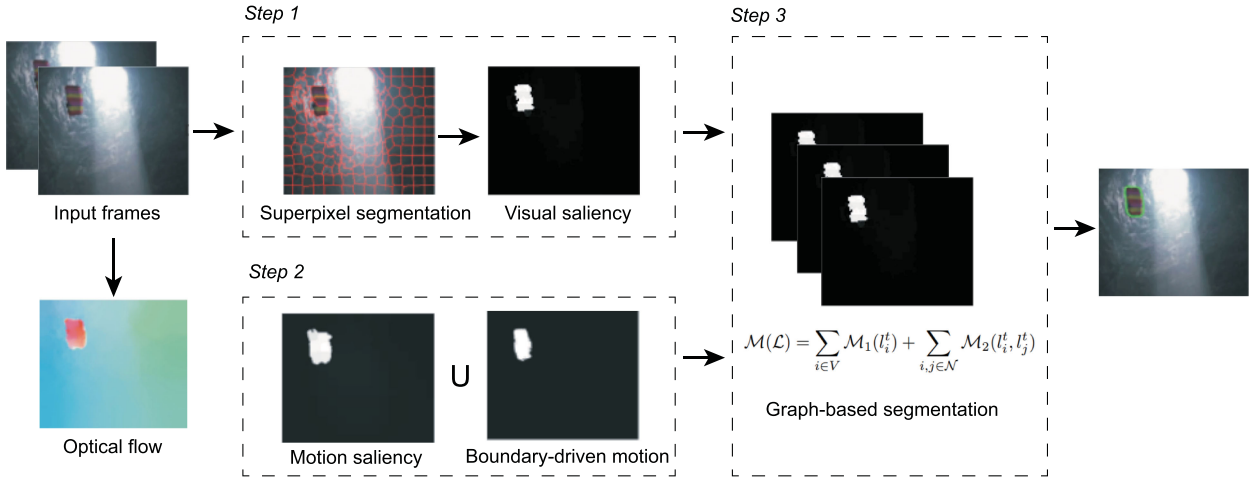


Fig. 2. The procedure of the proposed method.

graph with n nodes. Each node corresponds to the image location, i.e. image superpixel [42], and the edge link, denoted as w_{ij}, represents the similarity between the node pair (i, j). $\mathbf{W} = [w_{ij}]_{n \times n}$ is called the edge affinity matrix, and the degree matrix is $\mathbf{D} = diag\{d_{11}, \ldots, d_{nn}\}$, where $d_{ii} = \sum_j w_{ij}$. Manifold ranking is to compute the rank value f_i for each node in the graph with respect to a query y_i. In the following, we denote $\mathbf{f} = [f_1, \ldots, f_n]^T$. The optimal ranking of queries is to minimize the following energy function:

$$\sum_{ij} w_{ij} \left(\frac{f_i}{\sqrt{d_{ii}}} - \frac{f_j}{\sqrt{d_{jj}}} \right)^2 + \mu \sum_i (f_i - y_i)^2 \qquad (1)$$

where μ balances the smoothness constraint (i.e. the first term) and the fitting constraint (i.e. the second term). The first term ensures that the nearby nodes are assigned similar ranking scores, while the second term ensures that the predicted rank matches the query. The optimal solution is given in [43], i.e.

$$f^* = (\mathbf{I} - \alpha \text{Ç})^{-1} y_i \qquad (2)$$

where $\mathbf{I}$ is an identity matrix, $\text{Ç} = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$ is the normalized Laplacian of the graph and $\alpha = 1/(1+\mu)$. This method has been successfully utilized for image saliency detection in [44].

Graph-based Segmentation: The image segmentation problem can be posed as a binary labeling problem. Suppose that the given image is modeled as a graph, and each node corresponds to the image location (image pixel or image patch, etc.), as mentioned above. The labeling problem is to assign a label l_i for each node $i \in V$. That is, $l_i = 1$ denotes the foreground, while $l_i = 0$ means the background, and V is the set of all nodes. The solution $L = \{l_i\}$ can be formulated by the following function:

$$\mathcal{M}(L) = \sum_{i \in V} \mathcal{M}_1(l_i) + \sum_{i,j \in \mathcal{N}} \mathcal{M}_2(l_i, l_j) \qquad (3)$$

where $\mathcal{N}$ is the set of neighboring pixel pairs, $\mathcal{M}_1(l_i)$ is the data term, encoding the cost when the label of node i is l_i, and $\mathcal{M}_2(l_i, l_j)$ is the smoothness term, denoting the cost of labeling pixel pairs. This graph-based function can be solved efficiently via the graph cut [45], [46]. In this paper, we integrate the prior saliency information into it, and build the constraint model to boost the performance of the object segmentation in video.

III. The Proposed Method

To segment object automatically in video, we will describe the object via intra-frame and inter-frame analysis, respectively, in the video. The visual saliency computing is to find the salient object in the intra-frame, and the motion cues computing is able to localize the object in the inter-frames. Accordingly, it has three main steps: (1) visual saliency computing (Section III-A), (2) motion cues computing (Section III-B), and (3) video object segmentation (Section III-C). The procedure of the proposed method is illustrated in Fig. 2. Step 1 utilizes the proposed weighted multiple manifold ranking model to compute the visual saliency for each frame which provides valuable appearance prior for the likely foreground regions. Step 2 computes the motion cues as localization prior based on detection of the motion saliency and boundary-driven motion. Eventually, we define a new energy function for our task of video object segmentation based on the graph model in Step 3. Finally, the segmentation result is obtained after post-processing. In the following subsections, we will show the details of each step stated above.

A. Visual Saliency Computing

In [44], a ranking method that exploits the manifold structure data based on color feature for saliency detection is proposed. However, it is insufficient to distinguish a foreground object in video by a single feature. To circumvent this problem, we propose a weighted multiple manifold ranking method which can effectively combine different features to yield boosted performance for the visual saliency detection. The cost function $O(f)$ considers M manifolds each constructed by the different feature. The ranking score f can be obtained by minimizing the

cost function, i.e.

$$O(f) = \sum_{k=1}^{M} \left(\iota^k \sum_{i,j} w_{ij}^k \left(\frac{f_i}{\sqrt{d_{ii}^k}} - \frac{f_j}{\sqrt{d_{jj}^k}} \right)^2 \right) + \mu \sum_i (f_i - y_i)^2. \quad (4)$$

The first term ensures that nearby points are with similar ranking scores, while the second term ensures that the ranking score should fit the initial label value y_i. ι^k is the weight of the kth feature, and μ is the trade-off for the two terms. The minimum solution is computed by setting the derivative of the cost function to be zero. The ranking function is:

$$f = \left(\sum_{k=1}^{M} \iota^k (\mathbf{I} - \alpha \mathbf{C}^k) \right)^{-1} y_i, \quad (5)$$

where $\mathbf{I}$ is an identity matrix, $\mathbf{C}^k = \mathbf{D}^{k^{-1/2}} \mathbf{W}^k \mathbf{D}^{k^{-1/2}}$ is the normalized Laplacian matrix, and $\alpha = 1/(1+\mu)$. Also, we rewrite it as:

$$f = \left(\sum_{k=1}^{M} \iota^k (\mathbf{D}^k - \alpha \mathbf{W}^k) \right)^{-1} y_i, \quad (6)$$

where $\mathbf{W}^k = [w_{ij}^k]_{n \times n}$, and w_{ij}^k indicates the edge strength for each pair of nodes based on the kth feature, α is set at 0.99 to control the balance of two items in manifold ranking cost function, and μ is set at 0.01.

We first consider the color feature because it is one of the most important cues in the human vision system. We compute the average superpixel color and represent the color features using different color space representations, i.e. RGB, CIELab and HSV. Next, we utilize the global contrast and local contrast as color features. The global contrast of the ith superpixel is given by

$$DG_i = \Sigma_{j=1}^{N} d(c_i, c_j), \quad (7)$$

where $d(c_i, c_j)$ denotes the Euclidean distance between the ith and jth superpixel's average color c_i and c_j. The local contrast of color features is defined as

$$DL_i = \Sigma_{j=1}^{N} \lambda_{ij} d(c_i, c_j), \quad (8)$$

where $\lambda_{ij} = exp(-\frac{1}{\delta_c^2} \|p_i - p_j\|)$, p_i and p_j are the pixel positions and δ_c is empirically set at 0.3 thereinafter. Additionally, we utilize the histogram feature, denoted as

$$HD_i = \sum_{j=1}^{N} \sum_{k=1}^{q} \left[\frac{(h_{ik} - h_{jk})^2}{(h_{ik} + h_{jk})} \right], \quad (9)$$

where q is the number of histogram bins and we set q at 8. N is the number of superpixels. Finally, we adopt the histogram of gradient (HOG) [47] to describe the image features. It can be noted that the weight of the kth feature (ι^k) is set by its individual saliency detection performance using a small dataset.

The rank value in Eq.(6) indicates the relevance of a node to the background, and its complement is the saliency measure.

Fig. 3. The result of saliency detection, where the first column is the input image and the second column corresponds to its visual saliency.

Fig. 4. A snapshot of the results of motion saliency detection, where (a) the input frame, (b) its optical flow computed in the video sequences, and (c) detected motion saliency using the color feature.

The visual saliency using the normalized $\bar{f}$ (i.e. ranging between 0 and 1) is given as:

$$S(i) = 1 - \bar{f}(i), \quad (10)$$

where i indexes a superpixel node on graph. For instance, $w_{ij}^1 = e^{-\frac{\|c_i - c_j\|}{\delta^2}}$ can be defined as the edge strength using the superpixel's RGB color feature. An example of the saliency detection is illustrated in Fig. 3.

B. Motion Cues Computing

1) Motion Saliency Detection: We compute the optical flow for the video sequences using the state-of-the-art algorithm [48], which allows large displacements between frames and has efficient implementation (see Fig. 4(b)). In the optical flow field, the hue of a pixel indicates its direction and the color saturation corresponds to its velocity.

From the optical flow field, we can observe that the moving region usually has obvious color feature that makes it stand out from the background. Based on this property, we propose to

estimate the motion saliency score for each pixel in terms of the associated optical flow information. Based on the proposed algorithm in Eq.(6) and Eq.(10), we apply the color feature to calculate the motion saliency in the optical flow field. Hence, Eq.(6) can be simplified as

$$f = (\mathbf{D} - \alpha \mathbf{W})^{-1} y_i. \tag{11}$$

After getting the ranking score f, we can utilize Eq.(10) to compute the motion saliency for each pixel at each frame. A snapshot of the results of the motion saliency detection is shown in Fig. 4.

2) Boundary-driven Motion Detection: The above motion saliency can work well when either foreground or background exhibits dominant motion. However, in the real-world environment, objects in video sequences may demonstrate various motion states such as rotational motion, minor motion, or even static in some frames. To cope with these issues, we consider utilizing two different factors, i.e., motion boundary and static boundary, which are complementary and capable of providing strong cue for object parts of disparate appearance.

The first factor about the motion boundary is estimated by considering both the gradient and angle of a pixel using its optical flow information. Generally, the motion boundaries coincide with the object boundaries. Let $\vec{b}_i$ be the optical flow at pixel i. The strength of the motion boundary is defined as:

$$\mathcal{B}_i = \frac{\|\nabla \vec{b}_i\|}{exp(-\sum |\delta\theta_{i,j}|)} \tag{12}$$

where the numerator is the magnitude of the gradient of the optical flow for the pixel i, and the denominator denotes the aggregated angle difference between the pixel j and its neighbours with $\delta\theta_{i,j}$ indicating the angle difference of pixel i and j, and j is the neighborhood of i. The intuitive idea behind Eq.(12) is that a motion boundary pixel is prone to have much larger magnitude of gradient and different moving direction compared to its neighbours.

The second factor is to compute the static boundary for the potential objects in video sequences. Different from the traditional edge detection approaches, we derive the static boundary from the contours of superpixels with the help of the saliency in Eq.(10). The region (i.e. the group of superpixels) covering the thresholded saliency is selected, then the external boundary of these superpixels is detected as the static boundary for the foreground object.

Once the above two factors are computed, it can be regarded as boundary-driven cue to indicate the moving object. Then, we need to estimate which pixels are inside the object based on the point-polygon problem [49]. The idea is that any ray starting from a point inside the polygon will intersect the polygon boundary an odd number of times. We utilize the inside-outside algorithm [14] which provides an efficient implementation for this problem. An example of the localization prior for the motion region is illustrated in Fig. 5.

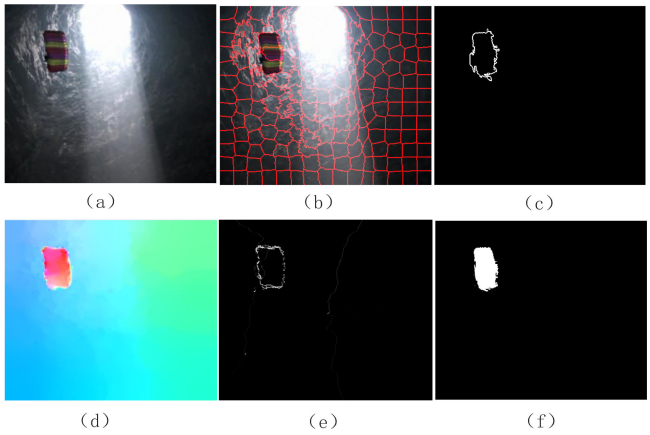


Fig. 5. The result of boundary-driven motion. (a) Input one frame. (b) Superpixels in the frame. (c) Static boundary of the likely foreground regions. (d) Optical flow of computed in the video sequences. (e) Motion boundary of the likely foreground regions. (f) Localization prior for the motion region.

C. Video Object Segmentation

After obtaining the visual saliency and motion cues in a video, the results are further refined by the graph-based method which formulates image segmentation as a pixel labeling problem. Each frame is divided to superpixels [42]. Each superpixel denoted as o_i^t can take a label $l_i^t \in \{0, 1\}$ indicating the background or foreground in the tth frame at the ith superpixel. It can be regarded as a node in the graph, thus we define the energy function for the labeling $\mathcal{L} = \{l_i^t\}_{i,t}$ of all superpixels at each frame in video, i.e.,

$$\mathcal{M}(\mathcal{L}) = \sum_{i \in V} \mathcal{M}_1(l_i^t) + \sum_{i,j \in \mathcal{N}} \mathcal{M}_2(l_i^t, l_j^t), \tag{13}$$

where V is the set of all pixels in the video, $\mathcal{N}$ consists of the neighboring superpixels, and i, j index the superpixels.

The data term $\mathcal{M}_1$ defines the cost of labeling superpixel i with label l_i at each frame. We utilize the Gaussian Mixture Model (GMM) based on the visual saliency to evaluate how likely a superpixel belongs to foreground or background. The data term is defined as:

$$\mathcal{M}_1(l_i^t) = -log(U_i^1(l_i^t) + \gamma \cdot U_i^2(l_i^t)), \tag{14}$$

where $U_i^1(l_i^t)$ is the pixel-likelihoods computed from each GMM of visual saliency. It means that a pixel that has similar visual saliency to the foreground (or background) will have high cost if labeled as the opposite value. At each frame t, we estimate a foreground GMM model from all superpixels with high visual saliency value (e.g. say over 0.7) in the video, weighted by how close in time they are to t. Then, U_i^1 is computed by the estimated GMM model. $U_i^2(l_i^t)$ is the localization prior derived from the motion cues, which provides a reliable information for the likely foreground objects in a frame, and $U_i^2 = f_{motion} + \phi(\mathcal{B}_m \cap \mathcal{B}_s)$, where f_{motion} is the motion saliency, B_m and B_s are motion boundary and static boundary, respectively. $\phi(\cdot)$ is a transformation generating the inside-outside map from boundary [14]. The parameter γ is a balancing constant. Similar to [14], we update $U_i^1(l_i^t)$ and $U_i^2(l_i^t)$ in the video sequences.

The smooth term $\mathcal{M}_2$ consists of two parts which encourage label smoothness in time and space, respectively. They are

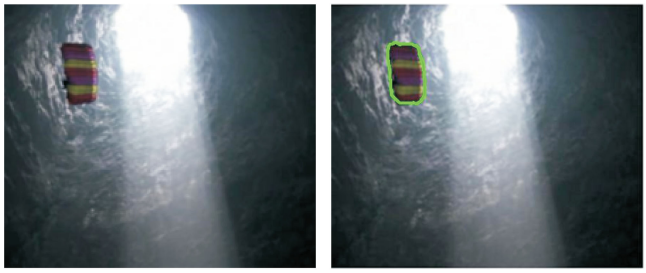


Fig. 6. A snapshot of the final segmentation result after post-processing.

given as:

$$\mathcal{A}^t_{i,j}(l^t_i, l^t_j) = \delta(l^t_i \neq l^t_j) e^{-\beta D_1(o^t_i, o^t_j) - D_2(o^t_i, o^t_j)} \quad (15)$$

$$\mathcal{H}^t_{i,j}(l^t_i, l^{t+1}_j) = \delta(l^t_i \neq l^t_j) e^{-\beta D_1(o^t_i, o^{t+1}_j) - D_2(o^t_i, o^{t+1}_j)} \quad (16)$$

where $\delta(\cdot)$ is the indicator taking values 0 or 1, $D_1(\cdot)$ is the color difference in RGB space, $D_2(\cdot)$ is the Euclidean distance two superpixels, and β is set as the averaged superpixels color difference of all pairs of neighboring superpixels.

Once we obtain the data term and smooth term for the labeling problem, we minimize Eq.(13) with the iterative graph cut [45], [46], [50], and use the resulting label (i.e. foreground or background) as the object segmentation at each frame, the foreground label is used to update the GMM model for image saliency in U^1_i and motion saliency in U^2_i. It runs several times until no foreground rebels change.

After obtaining the segmentation results, we can implement the post-processing to refine the object shape by utilizing the composition information of the image regions. Specifically, we run the forward step on all the video sequences, starting from the first frame towards the last one. We compare the object appearance in consecutive frames using patch-match algorithm [51] for quickly computing approximate nearest-neighbor fields between the pairs of image regions. Additionally, we utilize a mask transfer method [52] to automatically employ an adaptive stride size for mask transfer and interpolation estimating the lost parts for the object (mostly in case of occlusion), thus making the shape of object in the video sequences become more consistent. A snapshot of the final segmentation result is illustrated in Fig. 6.

To demonstrate the implementation for the salient object segmentation in the video, we summarize the whole procedure in Algorithm 1 **AVOS** as follows:

In the **AVOS**, Step 1 and Step 2 estimate the saliency-based multiple weighted manifold ranking and motion saliency, respectively. Step 3 adopts the graph model to label the foreground object and the background to obtain the salient object segmentation. The last step, i.e. Step 4, is a post-processing procedure to refine the final segmentation. The complexity of each step in **AVOS** algorithm is $O(n^2)$, $O(n)$, $O(n^2)$ and $O(n)$, respectively.

Algorithm 1: AVOS.

Input: Given the video sequences.
1: Calculate different salient features using Eq. (7-9) and combine them using Eq. (6); Estimate visual saliency in each frame using Eq. (10).
2: Compute the motion cues in the adjacent frames using motion saliency Eq. (11) and boundary-driven cue.
3: Segment the object using an iterative graph cut Eq. (13).
4: Refine the final segmentation through post-processing procedure.
Output: Object segmentation in each frame.

IV. Experimental Evaluation

A. Experimental Settings

In this section, we first compare our derived saliency to those produced by the other saliency detection methods. We utilize the *Precision*, *Recall* and*F-measure* as evaluation criteria to compare the performance with the existing counterparts. *Precision* is the ratio of correctly detected saliency region to the detected saliency region, while *Recall* is the ratio of correctly detected saliency region to the ground truth salient region. They are calculated as follows:

$$precision = \frac{\Sigma_i G_i B_i}{\Sigma_i B_i}$$
$$recall = \frac{\Sigma_i G_i B_i}{\Sigma G_i}, \quad (17)$$

where G_i and B_i are the value of the pixel i in the ground truth image G (i.e. the ground truth annotation of the image saliency), and binarized saliency image B (i.e. the binarized image of the detected saliency using adaptive threshold method), respectively. Additionally, *F-measure* is the overall performance measurement, which is computed as the weighted mean between the precision and recall values. It is given as:

$$F - measure = \frac{(1+\rho) \times precision \times recall}{\rho \times precision + recall}, \quad (18)$$

where ρ is a positive parameter to balance $precision$ over $recall$. A larger ρ will emphasize the precise detection of salient objects. In this paper, we set $\rho = 0.3$ to emphasize the precision.

Then, we evaluate the performance of the proposed AVOS on two video segmentation datasets: SegTrack dataset [7] and SegTrack v2 [53]. The performance of the video segmentation is measured by the mean intersection-over-union (mIoU) of the estimated segmentation and the ground truth across videos, i.e.,

$$mIoU = mean(\frac{\mathcal{L} \cap GT}{\mathcal{L} \cup GT}) \quad (19)$$

where $\mathcal{L}$ is the labeling result of the proposed method and GT is the ground-truth labeling in each frame in videos. In addition, the mean absolute error (MAE) is used as a complementary

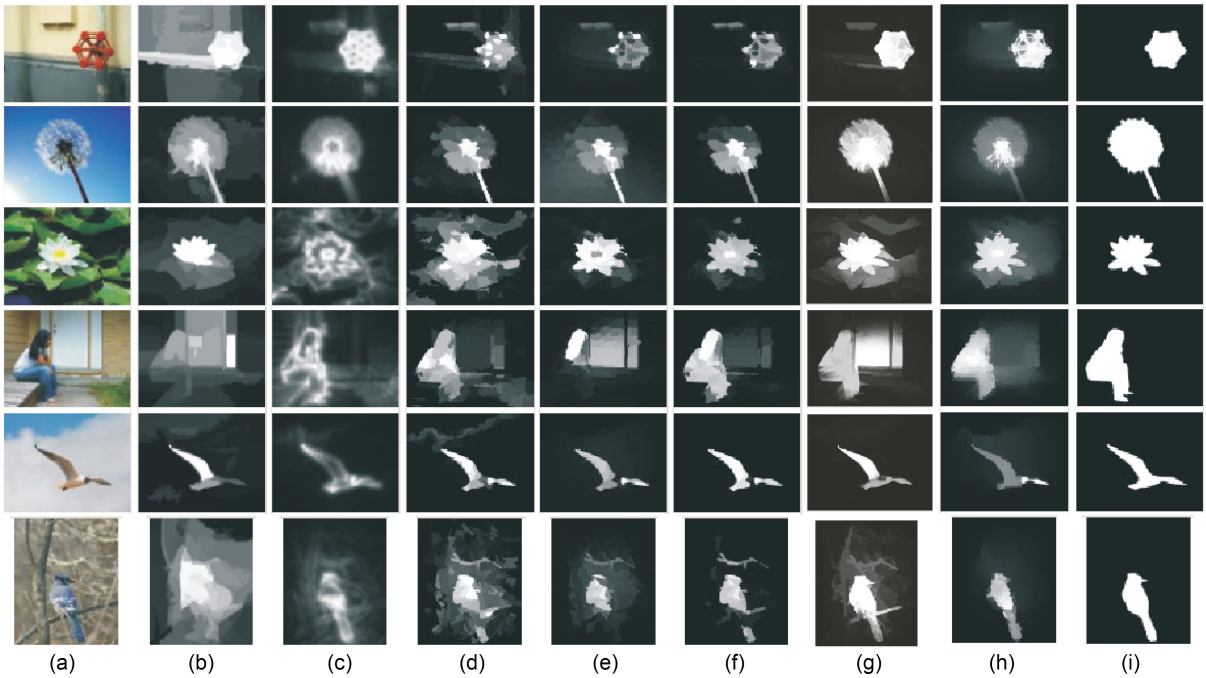


Fig. 7. The results produced by the different methods, where (a) input images, (b) AS [56], (c) CS [58], (d) GS [57], (e) MR [44], (f) SO [55], (g) MBS [59], (h) the proposed method, and (i) ground truth.

measure. MAE is defined as the average per-pixel difference between the extracted object map and the ground truth.

In the experiments, we use [42] to generate approximately 200 superpixels for each frame. To compute $U_i^1(l_i^t)$ in Eq.(14), we utilize 8 component GMMs, and γ in Eq.(14) is set at 0.75 by a rule of thumb. The proposed algorithm is implemented in MATLAB, with C/C++ implementations for critical functions. All the experiments are executed on an Intel i7, 3.4 GHz processor with 8 GB RAM.

B. Comparison Among Saliency Detection Algorithms

The proposed method can not only detect saliency in images, but also extract the salient objects in videos. We first perform the experiment on the MSRA-1000 dataset, which consists of 1,000 images with accurate human-labelled masks for salient objects [54]. The proposed method is compared with five state-of-the-art saliency detection methods: Zhu *et al.* (SO) [55], Yang *et al.* (MR) [44], Jiang *et al.* (AS) [56], Wei *et al.* (GS) [57], Goferman *et al.* (CS) [58] and Zhang *et al.* (MBS) [59]. The SO method utilizes an optimization framework to combine multiple low level cues to generate the saliency map, and the GS method adopts the boundary priors. The MR method is more related to ours, but it computes the saliency based on color feature only. Furthermore, the AS method detects the salient objects on multiple scales of the context. The CS method is a classical approach that utilizes the local and global factors and associated them with the high-level visual patterns to obtain image saliency. Differing from the works mentioned above, the proposed method efficiently and effectively incorporates various salient features with the different weights to generate the image saliency based on the manifold ranking.

Fig. 7 shows a snapshot of the image visual saliency using the different methods, where brighter pixels indicate higher saliency probabilities. It can be seen that the proposed method was capable of extracting visual salient regions, even for images with complicated background (e.g. bird). Overall, the estimated saliency by the proposed method provides more clear shape and appearance information which is beneficial for inferring the region of foreground object in the video sequences. Thus, it would bring less incorrect prior to the segmentation procedures. Later, we will verify that the use of our derived visual saliency along with motion cues would produce promising segmentation results.

The resulting *Precision*, *Recall* and *F-measure* are shown in Fig. 8, which provides a reliable comparison of how well various visual saliency highlight salient regions in the images. The proposed saliency method achieves the best performance up to a precision rate above 90%, which indicates our saliency maps are more precise and responsive to the foreground object in images.

C. Salient Object Segmentation

The proposed method is able to produce object segmentation results for video sequences in a fully unsupervised way. We compare the proposed method with the existing four competing ones, which are the most closely related works published in recent years. That is, we first consider the approach of proposed in [21], which aims at discovering the key segments across video frames as foreground objects using multiple appearance and motion cues. Then, we compare the most recently proposed

TABLE I
COMPARISON RESULTS BETWEEN THE PROPOSED METHOD AND THE FOUR COMPETING ALGORITHMS ON TWO DATASETS IN TERMS OF MIOU

	Ours	KEY [21]	NLC [16]	SAL [15]	OF [24]	CVS [34]	SVC [25]
birdfall	0.58	0.49	0.74	0.42	0.57	0.48	0.56
cheetah	0.56	0.45	0.69	0.33	0.34	0.36	0.40
girl	0.88	0.87	0.91	0.74	0.88	0.86	0.87
monkeydog	0.75	0.74	0.78	0.60	0.54	0.61	0.72
parachute	0.94	0.96	0.94	0.91	0.95	0.92	0.94

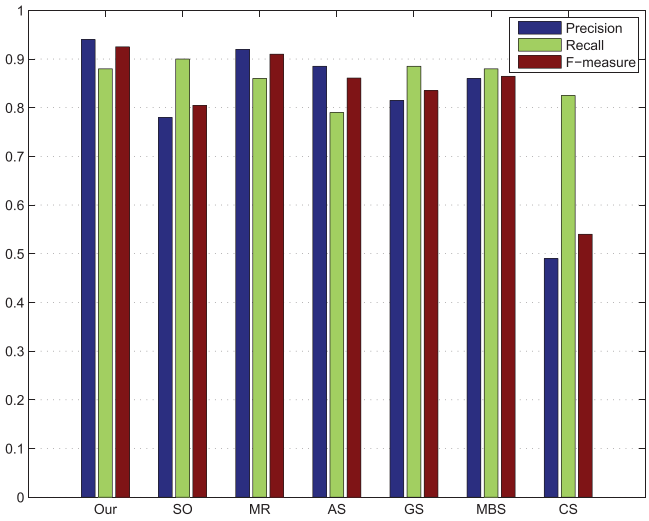


Fig. 8. The performance of seven methods measured by *Precision*, *Recall* and F-measure.

method which jointly optimizes optical flow and video segmentation [24] and one appearance model [25]. Finally, we compare the proposed method to three saliency-based approaches in [16], [15] and [34], respectively.

In this section, we first evaluated the proposed method on a small dataset, named Segtrack [7], which consists of six videos (*birdfall, cheetah, girl, parachute, monkeydog, penguin*) and the segmentation ground-truth for each video is available. The videos offer various challenges such as large camera motion (*girl*), large shape deformation (*monkeydog*) and low-contrast between the background and foreground (*cheetah*). Following [21], we discarded the penguin video because only a single penguin is labeled as the foreground amidst a group of penguins.

The comparison results between the proposed method and the above four competing ones are shown in Table I. It can be seen that the performance of the proposed method is comparable on most of the video sequences. The proposed method can handle videos captured by freely moving camera (e.g. *girl*), or with shape deformation motion (e.g. *monkeydog*), and videos with low contrast (e.g. *cheetah*). It outperforms the other methods because it is able to obtain better boundary for video object.

As for the NLC method [16], its random-walk transition matrix is robust against fast motion and achieved promising results. The method [24] also had competent performance. It builds a multi-level spatial-temporal graphical model with the use of optical flow and supervoxels, and jointly optimizes the model. Thus, both the segmentation and optical flow results can be improved by iteratively updating both models. Nevertheless, if the object mask for all frames is not accurate, the objects cannot be segmented correctly in the whole frames. Hence, the estimation of object mask in their method needs to be improved.

Furthermore, it is observed that the method of [21] tends to estimate the objectness for ranking their image segmentation results. This approach had high mis-segmentation rates for the sequence *cheetah* because it detected the background region as the foreground. In [15], it focuses on the estimation of spatiotemporal saliency and global appearance for the foreground object, and the results can be improved by comprehensive utilization of visual features, e.g. textures and shapes. The paper [25] needs the labeling work before the object segmentation, but the result is also favorable.

Fig. 9 shows a snapshot of video frames. Our method accurately segments the foreground object in these videos. In general, compared with the results from [15], [16], [21], [24], [25], [34], the proposed method performs well on this dataset. It is able to extract the foreground object effectively regardless of its various motion, and the segmented object demonstrates good boundary and appearance.

In the experiments, it is found that the motion cues play an important role in detecting of foreground objects in a video, especially when the visual saliency in the frame is not obvious. This is true for the *birdfall* sequences, in which the foreground object has prominent motion patterns compared to its surrounding while the saliency difference is insufficient. Fortunately, the motion cues of the *bird* in the video provide strong prior for its identifying. In the *girl* sequences, the visual saliency is clear and provides the appearance of the *girl*, but the motion cues may result in some obscure shape for the *girl* due to the inaccurate optical flow. However, by combining the visual saliency with motion cues, the proposed model can produce the satisfying result. Based on the prior analysis, we can have a conjecture that neither of these cues alone does not suffice to generate good segmentation results. Although the motion cues are able to give effective guidance for inferring the foreground object in the video, it is unwise to excessively dependent on them. We should utilize various features in spatial and temporal space, which usually provide more promising results.

We further carried out experiments on SegTrack v2 [53] and DAVIS [60] which are two more larger video segmentation dataset with full pixel-level annotations at each frame within each video. We compared our method with [15], [16], [21], [24], [25], [34] as well. The mIoU rates and MAE for different methods are shown in Table II.

The results in [15], [21] are not good because the foreground objects in some sequences have low contrast with the background or variable shape. The models are unable to obtain correct

Fig. 9. A snapshot of segmented objects using the different methods on video frames in SegTrack dataset. (a) Input video frames, (b) KEY [21], (c) NLC [16], (d) SAL [15], (e) OF [24], (f) CVS [34], (g) SVC [25], and (h) our method.

TABLE II
COMPARISON RESULTS OF THE PROPOSED METHOD AND THE FOUR COMPETING ONES ON SEGTRACK V2 DATASET IN TERMS OF MIOU AND MAE

Methods	SegTrack v2		DAVIS	
	mIoU	MAE	mIoU	MAE
KEY [21]	0.45	0.28	0.57	0.24
NLC [16]	0.80	0.11	0.64	0.19
SAL [15]	0.52	0.22	0.43	0.27
OF [24]	0.74	0.15	0.60	0.21
CVS [34]	0.56	0.23	0.51	0.26
SVC [25]	0.71	0.15	0.66	0.16
Ours	0.76	0.13	0.65	0.17

segments which are crucial for reliable estimation (e.g. the appearance estimation). The work [16] achieves the promising results, but it may fail in discovering the foreground objects when handling object that has similar appearance with background. The recent approach [24] obtains impressive segmentation results on these datasets. It works well under the condition of good optical flow estimation. Nevertheless, it is usually unable to segment the objects in case of the drift motion (e.g. 'drift-turn' video). In contrast, the proposed method can tolerate this by incorporating the visual saliency that does improve the quality of foreground object estimation.

We show the segmentation results obtained from the different approaches in Fig. 10. Furthermore, we demonstrate more examples to extract the different objects in varied scenes in Fig. 11. It should be noted that the proposed method would fail to work when part of the foreground object is more salient because the estimation of the visual saliency may be incorrect, as shown in Fig. 12.

D. Discussion

The visual saliency and the motion are two key factors that contribute to a good segmentation of the objects in the video. It is observed that the saliency tends to treat foreground as a whole,

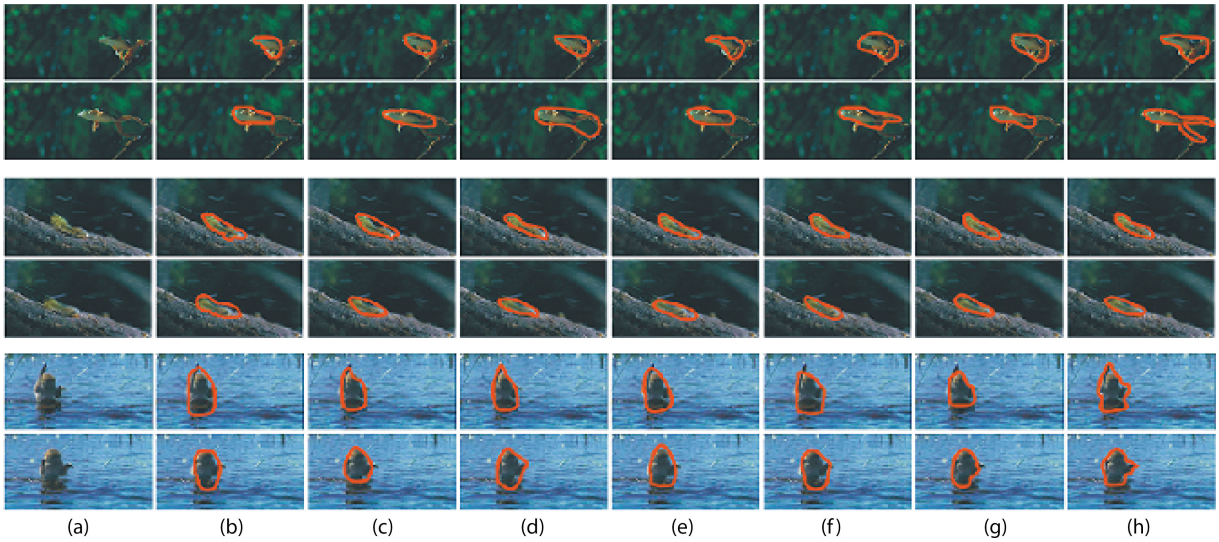


Fig. 10. A snapshot of the segmented objects using the different methods on video frames in SegTrack v2 and DAVIS datasets. (a) Input video frames, (b) KEY [21], (c) NLC [16], (d) SAL [15], (e) OF [24], (f) CVS [34], (g) SVC [25], and (h) our method.

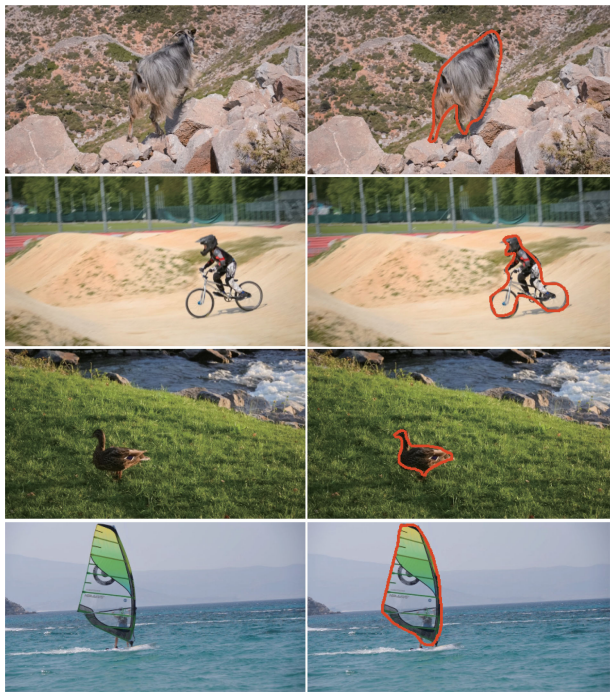


Fig. 11. A snapshot of the segmented objects on video frames

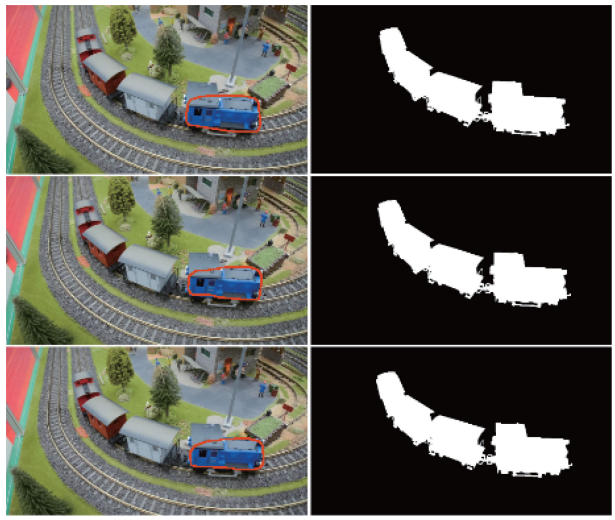


Fig. 12. One failure case with the proposed method, where the left column is the segmented object (marked in orange) and the right column is the ground truth.

and provides preliminary prior information for the object, such as shape and appearance. Thus, it will be very useful for unsupervised methods to detect the foreground object properly. When facing the challenging foreground and background, it might require one to observe both visual and motion cues together. In such cases, improved segmentation result is expected by utilizing the trajectory information of the extracted visual saliency and motion cues to determine the true foreground object. Compared with the methods [15], [16], the proposed method can generate much better boundary for the object, and the estimation of motion cues is more robust as well. This superiority is important for the object segmentation in various scenes in videos (e.g. in DAVIS dataset).

As for the computation time of the proposed method in comparison with the counterparts, Table III shows the average running time of the different methods running at a machine with an Intel Dual Core i7-3770 3.40 GHz CPU. It can be seen that the proposed method is still acceptable and would be further accelerated if GPU technique is used, which is, however, beyond the scope of this paper. We will therefore leave this issue elsewhere in our future studies.

TABLE III
AVERAGE RUNNING TIME OF THE PROPOSED APPROACH AND ITS COUNTERPARTS (SECONDS PER IMAGES)

Method	KEY [21]	NLC [16]	SAL [15]	OF [24]	CVS [34]	SVC [25]	Ours
Time (s)	20	12	8	32	2	5	30

V. CONCLUSION

In this paper, we have presented a new approach to automatic video object segmentation based on visual and motion saliency. Given a video sequence, the proposed method first computes the visual saliency to identify object-like regions in each frame by the proposed weighted multiple manifold ranking algorithm which combines different features to distinguish the foreground object from the background. We then compute motion cues to estimate the motion saliency and localization prior. Finally, to extract the salient object across all frames, we have designed a new energy function consisting of two terms. The first one is the data term depending on the visual saliency and localization priors, while the second term is the smoothness term depending on the time-space constraint. The experimental results have shown the effectiveness and robustness of the proposed method on benchmark datasets in comparison with the existing counterparts.

REFERENCES

[1] C.-W. Ngo, Y.-F. Ma, and H.-J. Zhang, "Video summarization and scene detection by graph modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 2, pp. 296–305, Feb. 2005.

[2] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *Proc. 10th IEEE Int. Conf. Comput. Vision*, 2005, pp. 1395–1402.

[3] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 12, pp. 1349–1380, Dec. 2000.

[4] W.-T. Li, H.-S. Chang, K.-C. Lien, H.-T. Chang, and Y. Wang, "Exploring visual and motion saliency for automatic video object extraction," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2600–2610, Jul. 2013.

[5] X. Bai, J. Wang, D. Simons, and G. Sapiro, "Video snapcut: Robust video object cutout using localized classifiers," *ACM Trans. Graph.*, vol. 28, no. 3, 2009, Art. no. 70.

[6] B. L. Price, B. S. Morse, and S. Cohen, "Livecut: Learning-based interactive video segmentation by evaluation of multiple propagated cues," in *Proc. 12th Int. Conf. Comput. Vision*, 2009, pp. 779–786.

[7] D. Tsai, M. Flagg, A. Nakazawa, and J. M. Rehg, "Motion coherent tracking using multi-label mrf optimization," *Int. J. Comput. Vision*, vol. 100, no. 2, pp. 190–202, 2012.

[8] J. Wang, P. Bhat, R. A. Colburn, M. Agrawala, and M. F. Cohen, "Interactive video cutout," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 585–594, 2005.

[9] X. Bai and G. Sapiro, "Geodesic matting: A framework for fast interactive image and video segmentation and matting," *Int. J. Comput. Vision*, vol. 82, no. 2, pp. 113–132, 2009.

[10] Y. Zhang, X. Chen, J. Li, C. Wang, and C. Xia, "Semantic object segmentation via detection in weakly labeled video," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 3641–3649.

[11] T. Wang and J. Collomosse, "Probabilistic motion diffusion of labeling priors for coherent video segmentation," *IEEE Trans. Multimedia*, vol. 14, no. 2, pp. 389–400, Apr. 2012.

[12] F.-C. Cheng, S.-C. Huang, and S.-J. Ruan, "Advanced background subtraction approach using Laplacian distribution model," in *Proc. IEEE Int. Conf. Multimedia Expo.*, 2010, pp. 754–759.

[13] J. Zhong and S. Sclaroff, "Segmenting foreground objects from a dynamic textured background via a robust Kalman filter," in *Proc. IEEE Int. Conf. Comput. Vision*, 2003, pp. 44–50.

[14] A. Papazoglou and V. Ferrari, "Fast object segmentation in unconstrained video," in *Proc. IEEE Int. Conf. Comput. Vision*, 2013, pp. 1777–1784.

[15] W. Wang, J. Shen, and F. Porikli, "Saliency-aware geodesic video object segmentation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 3395–3402.

[16] A. Faktor and M. Irani, "Video segmentation by non-local consensus voting," *Brit. Mach. Vision Conf.*, vol. 2, no. 7, 2014, Art. no. 8.

[17] J. Yang, B. Price, X. Shen, Z. Lin, and J. Yuan, "Fast appearance modeling for automatic primary video object segmentation," *IEEE Trans. Image Process.*, vol. 25, no. 2, pp. 503–515, Feb. 2016.

[18] K. Fragkiadaki, P. Arbelaez, P. Felsen, and J. Malik, "Learning to segment moving objects in videos," in *Proc. IEEE Conf. Comput. Vision Pattern Recogn.*, 2015, pp. 4083–4090.

[19] D. Zhang, O. Javed, and M. Shah, "Video object segmentation through spatially accurate and temporally dense extraction of primary object regions," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2013, pp. 628–635.

[20] T. Ma and L. J. Latecki, "Maximum weight cliques with mutex constraints for video object segmentation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2012, pp. 670–677.

[21] Y. J. Lee, J. Kim, and K. Grauman, "Key-segments for video object segmentation," in *Proc. IEEE Int. Conf. Comput. Vision*, 2011, pp. 1995–2002.

[22] S. A. Ramakanth and R. V. Babu, "Seamseg: Video object segmentation using patch seams," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2014, pp. 376–383.

[23] L. Chen, J. Shen, W. Wang, and B. Ni, "Video object segmentation via dense trajectories," *IEEE Trans. Multimedia*, vol. 17, no. 12, pp. 2225–2234, Dec. 2015.

[24] Y.-H. Tsai, M.-H. Yang, and M. J. Black, "Video segmentation via object flow," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 3899–3908.

[25] W. Wang, J. Shen, and F. Porikli, "Selective video object cutout," *IEEE Trans. Image Process.*, vol. 26, no. 12, pp. 5645–5655, Dec. 2017.

[26] T. Brox and J. Malik, "Object segmentation by long term analysis of point trajectories," in *Proc. IEEE Conf. Eur. Conf. Comput. Vision*, 2010, pp. 282–295.

[27] G. Evangelopoulos *et al.*, "Multimodal saliency and fusion for movie summarization based on aural, visual, and textual attention," *IEEE Trans. Multimedia*, vol. 15, no. 7, pp. 1553–1568, Nov. 2013.

[28] A. Oikonomopoulos, I. Patras, and M. Pantic, "Spatiotemporal salient points for visual recognition of human actions," *IEEE Trans. Syst., Man, Cybern. B Cybern.*, vol. 36, no. 3, pp. 710–719, Jun. 2005.

[29] Y. Yu, G. K. Mann, and R. G. Gosine, "An object-based visual attention model for robotic applications," *IEEE Trans. Syst., Man Cybern. B Cybern.*, vol. 40, no. 5, pp. 1398–1412, Oct. 2010.

[30] V. Gopalakrishnan, Y. Hu, and D. Rajan, "Salient region detection by modeling distributions of color and orientation," *IEEE Trans. Multimedia*, vol. 11, no. 5, pp. 892–905, Aug. 2009.

[31] K. Jerripothula, J. Cai, and J. Yuan, "Image co-segmentation via saliency co-fusion," *IEEE Trans. Multimedia*, vol. 18, no. 9, pp. 1896–1909, Sep. 2016.

[32] J. Lei *et al.*, "A universal framework for salient object detection," *IEEE Trans. Multimedia*, vol. 18, no. 9, pp. 1783–1795, Sep. 2016.

[33] W. Wang, J. Shen, X. Li, and F. Porikli, "Robust video object cosegmentation," *IEEE Trans. Image Process.*, vol. 24, no. 10, pp. 3137–3148, Oct. 2015.

[34] W. Wang, J. Shen, and L. Shao, "Consistent video saliency using local gradient flow optimization and global refinement," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 4185–4196, Nov. 2015.

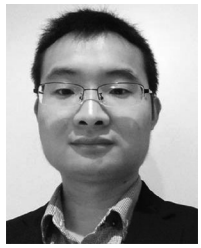
[35] W. Wang and J. Shen, "Video salient object detection via fully convolutional networks," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 38–49, Jan. 2018.

[36] W. Wang, J. Shen, Y. Yu, and K.-L. Ma, "Stereoscopic thumbnail creation via efficient stereo saliency detection," *IEEE Trans. Visualization Comput. Graph.*, vol. 23, no. 8, pp. 2014–2027, Aug. 2017.

[37] W. Wang, J. Shen, R. Yang, and F. Porikli, "Saliency-aware video object segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 20–33, Jan. 2018.

[38] W. Wang, J. Shen, J. Xie, and F. Porikli, "Super-trajectory for video segmentation," in *Proc. IEEE Int. Conf. Comput. Vision*, 2017, pp. 1680–1688.

[39] T. Vikrma, M. Tscherepanow, and B. Wrede, "A saliency map based on sampling an image into random rectangular regions of interest," *Pattern Recognit.*, vol. 45, pp. 3114–3124, 2012.
[40] C. Guo, Q. Ma, and L. Zhang, "Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform," in *Proc. IEEE Int. Conf. Comput. Vision Pattern Recognit.*, 2008.
[41] J. He, M. Li, H.-J. Zhang, H. Tong, and C. Zhang, "Manifold-ranking based image retrieval," in *Proc. 12th Annu. ACM Int. Conf. Multimedia*, 2004, pp. 9–16.
[42] A. Levinshtein *et al.*, "Turbopixels: Fast superpixels using geometric flows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 12, pp. 2290–2297, Dec. 2009.
[43] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf, "Ranking on data manifolds," *Adv. Neural Inf. Process. Syst.*, vol. 16, pp. 169–176, 2004.
[44] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2013, pp. 3166–3173.
[45] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001.
[46] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vision*, vol. 59, no. 2, pp. 167–181, 2004.
[47] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2005, pp. 886–893.
[48] N. Sundaram, T. Brox, and K. Keutzer, "Dense point trajectories by GPU-accelerated large displacement optical flow," in *Proc. Eur. Conf. Comput. Vision*, 2010, pp. 438–451.
[49] S. F. J. Foley, A. van Dam, and J. Hughes, *Computer Graphics: Principles and Practice*. Reading, MA, USA: Addison-Wesley, 1990.
[50] B. Peng, L. Zhang, and J. Yang, "Iterated graph cuts for image segmentation," in *Proc. Asian Conf. Comput. Vision*, 2009, pp. 677–686.
[51] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "Patchmatch: A randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 24–1, 2009.
[52] Q. Fan, F. Zhong, D. Lischinski, D. Cohen-Or, and B. Chen, "Jumpcut: Non-successive mask transfer and interpolation for video cutout." *ACM Trans. Graph.*, vol. 34, no. 6, pp. 195–1, 2015.
[53] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg, "Video segmentation by tracking many figure-ground segments," in *Proc. IEEE Int. Conf. Comput. Vision*, 2013, pp. 2192–2199.
[54] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2009, pp. 1597–1604.
[55] W. Zhu, S. Liang, Y. Wei, and J. Sun, "Saliency optimization from robust background detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2014, pp. 2814–2821.
[56] H. Jiang *et al.*, "Automatic salient object segmentation based on context and shape prior," in *Proc. Brit. Mach. Vision Conf.*, pp. 110.1–110.12, 2011, vol. 6.
[57] Y. Wei, F. Wen, W. Zhu, and J. Sun, "Geodesic saliency using background priors," in *Proc. Eur. Conf. Comput. Vision*, 2012, pp. 29–42.
[58] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2010, pp. 2376–2383.
[59] J. Zhang *et al.*, "Minimum barrier salient object detection at 80 fps," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 1404–1412.
[60] F. Perazzi *et al.*, "A benchmark dataset and evaluation methodology for video object segmentation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 724–732.



Qinmu Peng received the Ph.D. degree from the Department of computer Science, Hong Kong Baptist University, Hong Kong, in 2015. He is currently an Assistant Professor with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, Hubei, China. His current research interests include medical image processing, pattern recognition, machine earning, and computer vision.

Yiu-Ming Cheung (SM'06–F'18) received the Ph.D. degree from the Department of Computer Science and Engineering, Chinese University of Hong Kong, Hong Kong. He is currently a Full Professor with the Department of Computer Science, Hong Kong Baptist University, Hong Kong. His current research interests include machine learning, pattern recognition, visual computing, and optimization. He is the Founding Chair of the Computational Intelligence Chapter of the IEEE Hong Kong Section, and the Chair of the Technical Committee on Intelligent Informatics of the IEEE Computer Society. He serves as an Associate Editor of IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON PATTERN RECOGNITION, IEEE TRANSACTIONS ON KNOWLEDGE AND INFORMATION SYSTEMS, and *Neurocomputing*, to name a few. He is an IEEE Fellow, IET Fellow, BCS Fellow, RSA Fellow, and Distinguished Fellow of IETI.