Neurocomputing 172 (2016) 235–243

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Unsupervised pixel-level video foreground object segmentation via shortest path algorithm

Xiaochun Cao [a], Feng Wang [b,*], Bao Zhang [c], Huazhu Fu [d], Chao Li [e]

[a] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
[b] School of Computer Software, Tianjin University, Tianjin 300072, China
[c] School of Computer Science and Technology, Tianjin University, Tianjin 300072, China
[d] School of Computer Engineering, Nanyang Technological University, Singapore
[e] Shenzhen Key Laboratory of Data Vitalization, Research Institute of Beihang University in Shenzhen, Shenzhen 518057, China

ARTICLE INFO

Article history:
Received 1 November 2013
Received in revised form
31 July 2014
Accepted 20 December 2014
Available online 9 May 2015

Keywords:
Video object segmentation
Shortest path solution

ABSTRACT

Unsupervised video object segmentation is to automatically segment the foreground object in the video without any prior knowledge. In this paper, we propose an object-level method to extract the foreground object in the video. We firstly generate all the object-like regions as the segmentation candidates. Then based on the corresponding map between the successive frames, the video segmentation problem is converted to corresponding graph model, which selects the most corresponding object region from each frame. The shortest path algorithm is explored to get a global optimum solution for this graph. To obtain a better result, we also introduce a global foreground model to restrict the selected candidates. Finally, we utilize the selected candidates to obtain a more precise pixel-level foreground object segmentation. Compared with the state-of-the-art object-level methods, our method does not only guarantee the continuity of segmentation result, but also works well even under the cases of fast motion and occlusion.

1. Introduction

Social media contains a large scale videos. How to extract the main contents for these videos automatically is a key problem for media analytics and learning. A common media issue is the video foreground segmentation. Video object segmentation methods generally include two categories: supervised segmentation and unsupervised segmentation. Supervised segmentation needs user interaction to initialize the key objects. In contrast, the unsupervised methods release the user input to automatically extract the foreground. There is relatively less attention focusing on the unsupervised method compared with the supervised one because it is difficult to define the foreground in video automatically. But the study of unsupervised video object segmentation is significant and it can be applied to many fields such as video analysis and understanding [1], video summarization and indexing [2,3], video retrieval [4], and activity understanding [5,6]. For instance, object-based video segmentation separates the meaningful object from the background. Video analysis and understanding can have a better understanding of foreground and background, as well as the semantic relationship of them. Moreover, for video retrieval, object segmentation can find the videos with related foreground, and remove unrelated videos which have the similar background.

Recently, there are many methods provided to predict foreground model from videos. For instance, visual saliency has been used to form foreground object model in the video [7–10]. Gu et al. [11] and Wang et al. [12] propose video scene segmentation approaches with content coherence and contextual dissimilarity. Brox and Malik [13] propose a method based on foreground object trajectory. A series of trajectories are firstly extracted from the video, then processed by the spectral clustering method. Since plenty of the trajectories are sparse, the calculated foreground also consists of sparse pixels set. Ochs and Brox [14] extend the method [13] to acquire a more dense foreground region. Brendel and Todorovic [15] argue that video object segmentation by tracking regions has many fundamental advantages over the approaches based on tracking points or jointly clustering of all pixels from all video frames. However, Lee et al. [16] point out that these methods lack an explicit notion of what a foreground object should look like in video data and the low-level grouping of pixels usually results in over-segmentation.

Since the above methods are based on the low-level visual features, which lack an explicit notion of what a foreground object should look like in video data, methods in [17–19,32,33] explore object-based segmentation in static image and achieve significant progress. Those methods generate multiple object hypotheses and rank hypotheses according to their scores. The model is learned for

* Corresponding author.
E-mail addresses: caoxiaochun@iie.ac.cn (X. Cao), wangf.tju@gmail.com (F. Wang), zhangbao@tju.edu.cn (B. Zhang), hzfu@ntu.edu.sg (H. Fu), licc@buaa.edu.cn (C. Li).

http://dx.doi.org/10.1016/j.neucom.2014.12.105
0925-2312/

a generic foreground object using several image features such as color, texture, and boundary, which is then object category independent.

In this paper, we propose an object-level method to segment the foreground object from the unlabelled. We firstly generate a set of the object-like regions by using method [17]. Then we construct the corresponding graph model based on the corresponding map using selected candidates between the successive frames. After that, the shortest path algorithm is explored to get a global optimum solution. To obtain a better result, we employ an interaction method that introduces a global model to restrict the selected candidates with the global object model. Finally, the selected candidates are utilized to obtain a more precise pixel-level foreground segmentation.

1.1. Related works

Lee et al. [16] introduce an unsupervised video object segmentation method based on object level. They firstly get a series of candidate regions [17]. Combined with the motion attribute, each proposal is scored. Then based on the color histogram of each proposal, different object clusters are acquired using spectral clustering. The cluster with the highest mean score is regarded as the foreground, which is used for video object segmentation. It is the common sense that the object going through the entire video always has more significance than the short-term appearing object, i.e. more likely to be the foreground object. However, there are three major problems when they utilize spectral clustering to form a foreground model. (1) The clustering is merely based on color histogram to measure the correspondence distance (correlation) between two proposals. Since no other features are taken into consideration to balance the correlation between two proposals in the successive frames, this might result in wrong classification when background color is similar to the foreground color. (2) They cannot guarantee that the proposals in their clustering result cover all the frames in the video, i.e. lack of continuity. (3) As they said, the cluster with highest score is selected to be the foreground object model. However, if the wrong foreground object is chosen (e.g. there is an interferent object with faster motion in the background, the interferent object could be selected to be the foreground object model), the final segmentation result would be wrong even after the post-processing steps. The comparison with Lee's method is shown in Fig. 1. We simulate a common scene happening all the time. It contains a walking people and a running one. In the first several frames, both people exist in the video. Since the running one is quicker to run out of the camera sight, the last few frames contain the walking person only. In contrast, our method works better than Lee's under such scene.

Ma and Latecki [20] attempt to address this video object segmentation problem by utilizing relationships between object proposals in adjacent frames. The object region candidates are selected simultaneously to construct a weighted region graph. This problem is modeled as finding a constrained Maximum Weight Cliques problem. However, this method also cannot guarantee that the proposals cover all the frames in the whole video. Moreover, the major problem is that the approach to solve maximum weight cliques is NP-hard. Only an approximate optimization solution is used to obtain the result, which may be not the global optimal solution. What is more, [16,20] have an additional limitations compared to the proposed method using object based segmentation approaches. The object proposal selection of a particular frame does not depend directly on adjacent frames in both approaches.

Zhang et al. [21] present a new approach to improve the methods [16,20], which uses a novel and efficient layered Directed Acyclic Graph (DAG) based approach to segment the primary object in videos, and the problem converts to find the highest weighted path in the DAG. The problem could be solved by dynamic programming in linear time. This approach also uses innovative mechanisms to compute the 'objectness' of a region and to compute similarity between object proposals across frames. However, the solving method of dynamic programming maybe causes curse of dimension, if the number of proposals and frames is large enough.

1.2. Our framework and contributions

In order to overcome the aforementioned drawbacks, we propose a new approach to segment the foreground object. Our method is to select object-like regions according to both static and dynamic cues, and then segment the main object of the unannotated video using

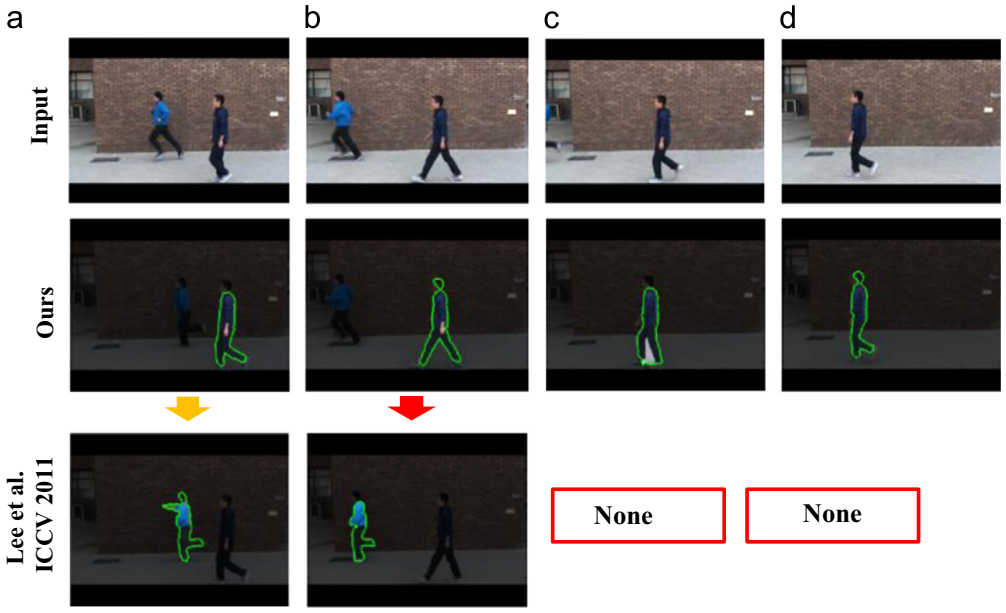


Fig. 1. Comparison with method [16]. The first row is input frames. Our result is shown in the second row, compared with the result of [16] shown in the last row. There is no corresponding segmentation result with respect to frames 9 and 13, marked by the red boxes. Note that the proposed method was able to find objects in all frames. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.) (a) Frame 1 (b) Frame 5 (c) Frame 9 (d) Frame 13.

those selected candidates. We formulate the candidate selection into a corresponding graph model problem. In this graph, the nodes are represented as the candidates, and the edges connect these candidates. Different from the common solutions in [16,20–22], we solve it using shortest path algorithm to get the global optimum solution. We also introduce a global model constraints to refine the selected candidates. After that, we use each selected candidates to define the foreground and background model, and obtain a pixel-wise segmentation using graph cuts.

In general, the main contributions of this paper are the following: (1) we introduce a novel method to provide a robust and continuous foreground model dealing with the video segmentation problem of the state-of-the-art which is the key problem for media processing, (2) a global optimum solution of video object segmentation based on shortest path algorithm is given the candidate foreground proposals, (3) to make the selection approach more precise, a global model is learned to restrict the selection of proposals in each frame, (4) a wide range of applications such as video retrieval and media analytics.

A preliminary version of this work was published at [23]. In this paper, we present a global model to restrict the selection of proposals in each frame. Furthermore, we proceed a more precise pixel-level foreground segmentation compared with [23].

2. Approach

Our goal is to get a global optimum solution of video object segmentation given the candidates. The proposed framework is shown in Fig. 2. There are five main steps to our approach: (a) scoring each video frame region (proposal) using appearance and motion cues to determine the probability of a foreground object using the method [17]; (b) calculating the correspondence value between two proposals in successive frames to measure the continuity of foreground object; (c) constructing the corresponding graph model and solving it using shortest path method; (d) learning a global model to restrict the selected candidates of each frame; and (e) utilizing the selected candidates to obtain a more precise pixel-level foreground segmentation. We now describe each step in turn.

2.1. Finding object-like region in video

Several methods [17,18] could generate the object-like regions in the frame without any prior knowledge about the foreground object. To get the proposals, we use these methods to generate K regions for each frame in the video and $K \times N$ regions in total for a video consisting of N frames. We initially generate object proposals of each frame [17] to generate the candidates which have a high score in terms of both appearance and motion may be more similar to a true object with these well-defined properties. The proposal scoring function $S(P)$ is defined as

$$S(P) = A(P) + M(P), \tag{1}$$

where P, $A(\cdot)$ and $M(\cdot)$ denote the proposal candidate, static intra-frame appearance score and dynamic inter-frame motion score, respectively. $A(P)$ is computed from [17]. It stands for the probability whether a real object is in this region. In order to compute this score, many static cues are used, such as the color differences with nearby pixels and occlusion boundaries. $M(P)$ is computed using optical flow histograms of both proposals and the bounding boxes around the proposals as in [16]:

$$M(P) = 1 - \exp(-\chi^2_{flow}(P, \overline{P})), \tag{2}$$

where optical flow histograms of the proposal P and the pixels $\overline{P}$ around it within a loosely fit bounding box are computed, and χ^2_{flow} is the χ^2-distance between optical flow histograms. Both $A(P)$ and $M(P)$ of all regions in the video are normalized using the distributions of scores.

2.2. Calculating correspondence value

To achieve the continuity between the successive frames, we should define similarity between two proposals. We utilize optical flow [24] instead of the commonly used color feature. Optical flow is more robust to measure the correspondence between two proposals in successive frames, when the background has a similar color with foreground object. Optical flow is the velocity field which warps one frame I_i into the next frame I_{i+1}, which means each pixel in frame I_i has a corresponding pixel in the next frame I_{i+1}. Thus the mth proposal P_i^m in frame I_i has a corresponding region $R(P_i^m)$ in the next frame I_{i+1}. The correspondence value is defined as

$$C(P_i^m, P_{i+1}^n) = \frac{|R(P_i^m) \cap P_{i+1}^n|}{|P_{i+1}^n|}, \tag{3}$$

where $|\cdot|$ denotes the pixel number in the proposal. Note that, there is often the case, when the corresponding region $R(P_i^m)$ of one proposal P_i^m in the ith frame covers the proposal P_{i+1}^n in the next frame, the calculated correspondence will be high. However, it is not a good match. To solve this problem, the reversed corresponding map $\overline{C} = C(P_{i+1}^n, P_i^m)$ is calculated. Considering the correspondences from both directions between two proposals in successive frames, the

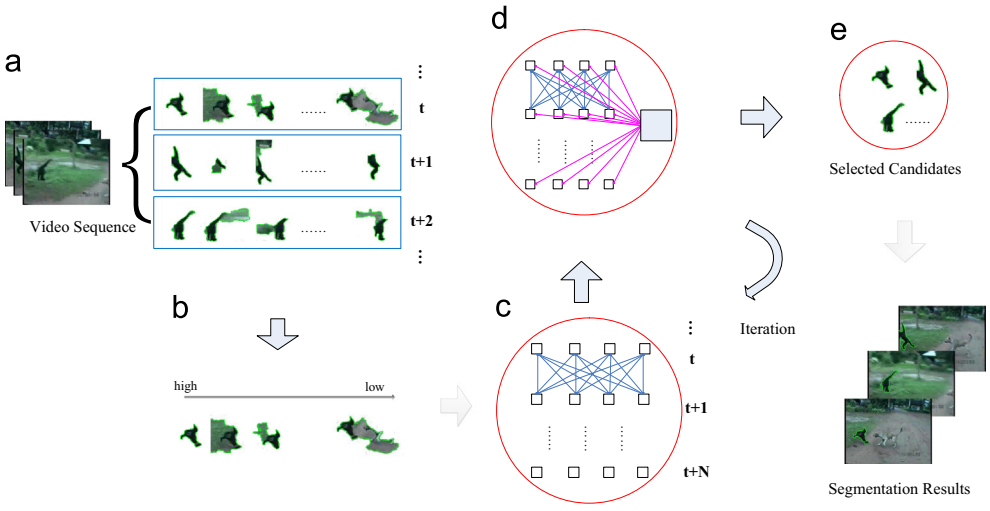


Fig. 2. Our video object segmentation framework (a) Object Candidate, (b) Calculating Correspondence Value, (c) Constructing the Graph, (d) Learning Global Models, (e) Pixel-level Foreground Object Segmentation.

corresponding map is generated and will be further illustrated in the following subsection.

2.3. Constructing the corresponding graph

Our goal is to segment the foreground object from the input video. From the above two subsections, each proposal score and correspondence value between two proposals in successive frames are defined. Therefore, we employ a corresponding graph to connect the candidates from the neighboring frames. In our graph, the nodes are candidates and the edges are the similarity between two candidates. We are given a video sequence containing N frames $V=\{I_1, I_2, \ldots, I_N\}$. $I_i, 1 \le i \le N$ denotes the ith frame. A proposal segmentation $P_i^j, 1 \le i \le N, 1 \le j \le n_i$, denotes the jth proposal in frame I_i. n_i is the proposal number in the frame I_i. P_i^j is a binary labeling assigning to each pixel in the frame I_i a label 0 for background, and label 1 for foreground. Since the foreground object is represented by one proposal, only one proposal is selected in each frame of the video. Each frame I_i is a node which must take on a state from a discrete set corresponding to all image proposals in this frame. The goal is to find a labeling $\mathrm{x} = \{x_i | i = 1, \ldots, N, 1 \le x_i \le n_i\}$ that minimizes the energy function:

$$E(x) = \sum_i \Phi(P_i^{x_i}) + \lambda \sum_{|i-j|=1} \Psi(P_i^{x_i}, P_j^{x_j}). \tag{4}$$

Herein, the parameter λ balances the unary term Φ and pairwise term Ψ.

The unary potential Φ measures how likely a proposal $P_i^{x_i}$ is to contain the foreground object. We have already known that high score $S(P_i^{x_i})$ denotes high probability being a foreground object as illustrated above. In this energy function, it is denoted as

$$\Phi(P_i^{x_i}) = 1/(1 + S(P_i^{x_i})), \tag{5}$$

The pairwise potential Ψ measures the similarity between two proposals $P_i^{x_i}$ and $P_j^{x_j}$ in the neighboring frames assessing how likely they are to contain objects of the same object. While a low value Ψ represents that two proposals are more likely to be the same object:

$$\Psi(P_i^{x_i}, P_j^{x_j}) = \exp(-\tfrac{1}{\mu}(C(P_i^{x_i}, P_j^{x_j}) \times C(P_j^{x_j}, P_i^{x_i}))), \tag{6}$$

where μ denotes the mean of all the values in Ψ. This equation rescales the corresponding map to the range $(0, 1]$. $C(P_i^{x_i}, P_j^{x_j})$ is the correspondence value in Eq. (3).

Normally, TRW-S [22] is utilized to solve this energy function. However, the result produced by TRW-S is local optimal solution. Therefore, it inspires us to convert this question to a particular graph model. Firstly, both proposals and correspondences between two proposals in the successive frames are considered as graph nodes. The proposal node value is the value of the unary term Φ and the correspondence node value is the value of the pairwise term Ψ. Assume S and T be the starting and end node, respectively. S is connected with all the proposals in the first frame, while T is connected with all the proposals in the last frame. Since correspondence only exists between the successive frames, the solution of Eq. (4) is the shortest path value from starting node S to end node T. The shortest path provide a global solution for candidates selection.

2.4. Learning global models

Due to using the unsupervised method to segment video object and having no prior knowledge on size, location, shape, or appearance of the object, some errors may exist in some frame, such as occlusion by other objects or disappearance from the scene. Therefore, the shortest path method may be not accurate, because the values in Eq. (1) are computed according to single frame instead of global.

To solve the problem, we add an interaction approach to refine the selected candidates which introduces a global constraint to avoid selecting bad result. The detail is that after constructing the corresponding graph, we use the shortest path method to select proposals in each frame. Then, we utilize the regions gained from the shortest path method as foreground to learn a global object model. Then we use this global model to calculate the each score of the $K \times N$ regions. Finally, we normalize the scores and add them to Eq. (1), as

$$S(P) = A(P) + M(P) + G(P), \tag{7}$$

where $G(\cdot)$ is the score calculated by the global model. Gaussian Mixture Models (GMM) model is employed to represent the global model. To make the result more precise, we apply this step recursively. The interaction is stopped when the results do not change, or it reaches the maximum interactive numbers. Fig. 3 shows the iterative example.

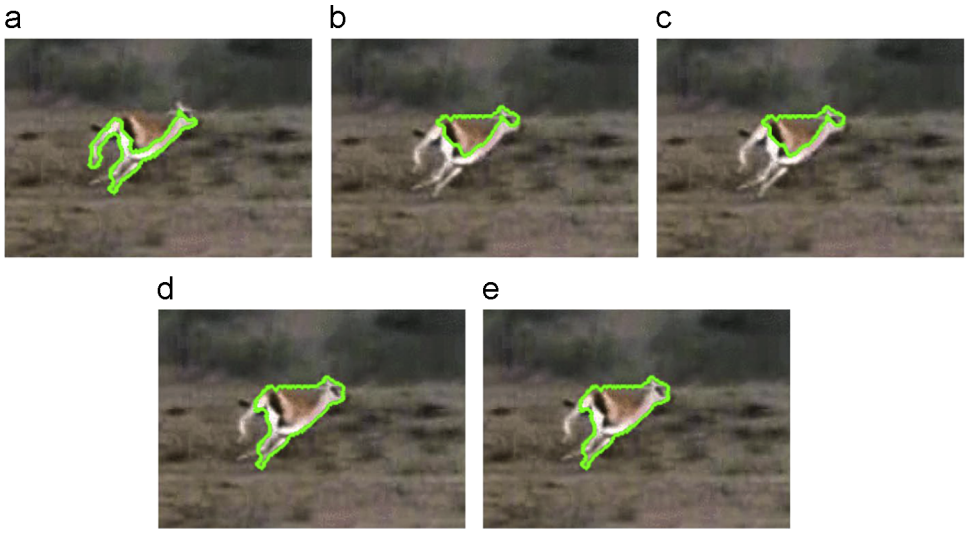


Fig. 3. The iterative example. The iterative number is 5. The result reflects that the proposed method of learning global model can improve the candidate selection (a) Iteration 1 (b) Iteration 2 (c) Iteration 3 (d) Iteration 4 (e) Iteration 5.

2.5. Pixel-level foreground object segmentation

We obtain the rough object segmentation of each frame in the video. However, the segmentation using the selected candidates generated by [17] is not very precise. Therefore, we can utilize the selected candidates to define a foreground and background model. We make use of color and location prior of exactly one object proposal in each frame.

We use all pixels in each frame to construct a corresponding graph, and the node and edge between two nodes denote the pixel and cost of a cut respectively. The energy function is defined as Eq. (8) so that minimization of it could correspond to a good segmentation:

$$E(f)=\sum_{i\in I}D_i(f_i)+\xi\sum_{i,j\in N}V_{i,j}(f_i,f_j), \tag{8}$$

where $f=\{f_1,\ldots,f_n\}$ denotes labels of each pixel, $f_i\in\{0,1\}$ with 0 for background and 1 for foreground and N consists of four spatially neighboring pixels in the same frame and two temporally neighboring pixels in adjacent frames. Optical flow vector displacement is used to assign the temporal neighbors of pixels in the next frame.

For the smoothness term $V_{i,j}$, we follow the function definitions in [25], which favors assigning the same label to neighboring pixels having similar color:

$$V_{i,j}(f_i,f_j)=[f_i\neq f_j]\exp^{-\beta(u_i-u_j)^2} \tag{9}$$

where $[\phi]$ denotes the indicator function taking values 0, 1 for a predicate ϕ, and u_i, u_j are the set of pairs of neighboring pixels. The constant β is chosen by [26] to be $\beta=(2\left\langle(u_i-u_j)^2\right\rangle)^{-1}$, where $\langle\cdot\rangle$ denotes expectation of the pixels.

For the data term D_i, we use the function defined in [16] which defines the cost of pixel i with label f_i:

$$D_i(f_i)=-\log(\alpha\cdot U_i^c(f_i)+(1-\alpha)\cdot U_i^l(f_i)) \tag{10}$$

where $U_i^c(\cdot)$ is the color-induced cost and $U_i^l(\cdot)$ is the local shape match-induced cost. To compute the $U_i^c(f_i)$, we first estimate two Gaussian Mixture Models (GMM) to model the foreground (*fg*) and background (*bg*) appearance. The proposals we selected by the method of shortest path can learn the *fg* model, and the complement of selected candidates in each frame can serve as the *bg* model. To compute the $U_i^l(f_i)$, we utilize technique in [16] to prime the location and scale of the foreground object in a frame using shapes [27] of the object candidates selected by shortest path method.

Finally, in order to segment the whole video, we minimize Eq. (8) with binary graph cuts [28] after obtaining the data term D and smoothness term V. Then we use the resulting label assignment as the foreground object segmentation of the video and thus get the final segmentation results.

As mentioned above, Fig. 2 shows the flowchart of our method, which has five steps. The first two steps generate object candidates and calculate candidates score. In the next, we construct the corresponding graph shown in Fig. 2(c). The shortest path method is utilized to compute the selected candidates. We employ all the candidates to form a global model, which makes the result more accurate shown in Fig. 2(d). Finally, we apply all candidates for the pixel-level foreground object segmentation. The detail computation of our algorithm is described in Algorithm 1.

Algorithm 1. Our algorithm.

Inputs: the total number of frames N, video sequence $V=\{I_1,I_2,\ldots,I_N\}$

1: Calculate object-like proposal $P_i{}^j$ in each frame I_i
2: Compute proposal score S_i^j for P_i^j
3: Compute correspondence value $C(P_i^m,P_{i+1}^n)$ between P_i^m and P_{i+1}^n
4: Construct the graph $G(V,S,C)$
5: $k=0$
6: **while** $k\leq$ maximum interactive numbers **do**
7: Select the candidates via the shortest path
8: Learn the foreground global models
9: Update proposal score S_i^j
10: $k=k+1$
11: **end while**
12: Obtain pixel-level foreground object segmentation

3. Experiments

To evaluate the performance of our method, we test our method on a variety of benchmark datasets. Qualitative and quantitative analyses of our results are presented.

3.1. Experimental Settings

(1) *Datasets*: We test our algorithm on three well-known segmentation datasets: the SegTrack dataset [29], Berkeley Motion Segmentation Dataset (BMSD) [13] and GaTech video segmentation dataset [30]. The first one is mainly used for segmentation and tracking, and the other two focus on segmentation. SegTrack Dataset [29] has six videos: monkeydog, bird, girl, birdfall, parachute, and penguin. For the six videos, a pixel-level ground-truth is provided for the primary foreground object. In the other dataset BMSD [13], there are four groups of 26 video sequences involving car (10), marple (13), people (2) and tennis (1). The GaTech video segmentation dataset [30] collects 15 video sequences, and the length of these videos is longer than the first two datasets.

(2) *Baselines*: In our experiment, we adopt five state-of-the-art methods for comparison, which are Zhang [21], Ma [20], Lee [16], Tsai [29] and Chockalingam [31]. The methods in [21,20,16] and our method are unsupervised. They automatically discover the primary object in image as well as segment the object out. The methods proposed by [29] and [31] are supervised which need label the first frame of each video.

(3) *Evaluation metrics*: To evaluate the effectiveness of our method, we test our results based on the segmentation error as measured by the average number of incorrect pixels per frame compared to the ground-truth. Specially, it is computed as [29]

$$error=\frac{XOR(f,GT)}{F} \tag{11}$$

where f is the calculated label for every pixel of the method result, GT is the ground-truth label of the input video, and F is the number of frames in a given video. Since all videos are roughly of the same size, the average error rate over the 5 videos is computed as average over all frames in all videos, i.e., we treat all 5 videos as a single video and apply Eq. (11).

3.2. Quantitative analysis

We evaluate our algorithm on SegTrack dataset which provides ground-truth for the primary foreground object. In the SegTrack dataset, there are six videos. Same as [16], we do not evaluate our method on penguin video since only a single penguin is labeled as the foreground object among a group of penguins. We show other four segmentation results of them in Fig. 4. As we know, object segmentation in these videos is extremely challenging due to several facts, such as the primary object is with large shape changes, foreground and background color are similar. For each video, we select 4 frames to show the result. Taking cheetah as an

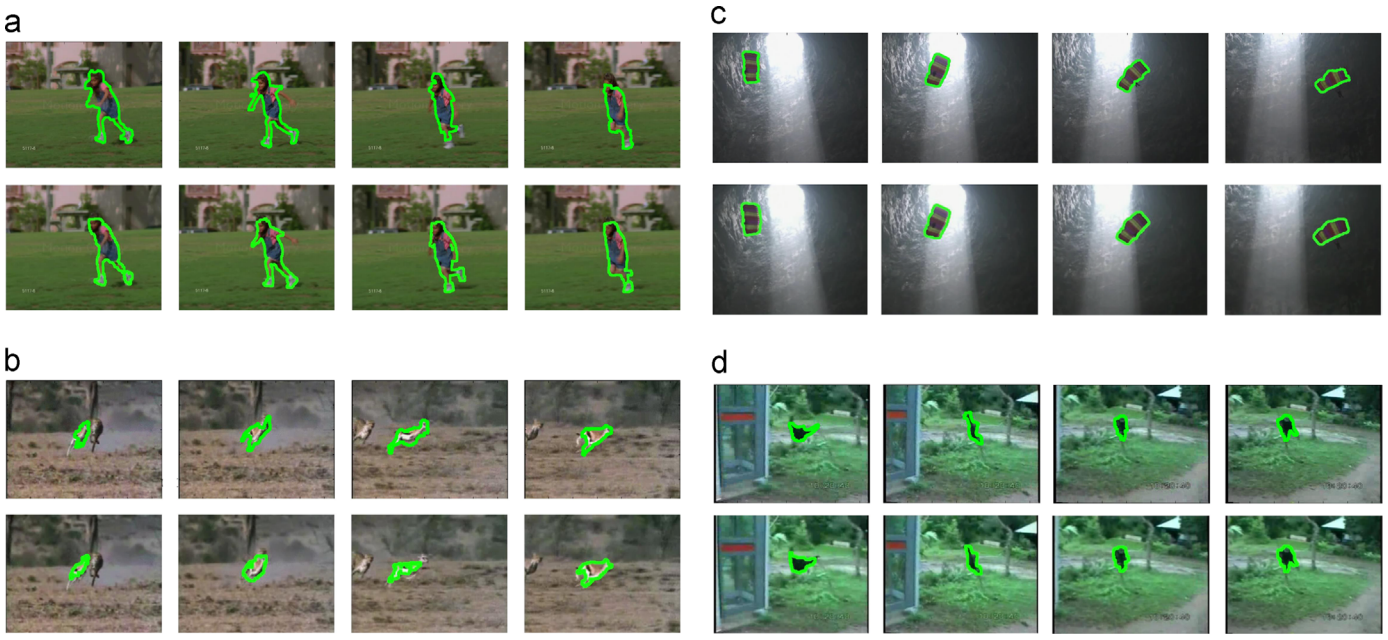


Fig. 4. SegTrack dataset results. Segmented primary foreground objects are marked by green boundaries. Our result is shown in the first row, compared with Lee [16] result shown in the second row. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.) (a) girl (b) cheetah (c) parachute (d) monkeydog.

example shown in the Fig. 4(c), we firstly calculate all the proposal and correspondence values for the video sequence. Then we use shortest path algorithm and proceed pixel-level foreground object segmentation. Obviously, one cheetah is segmented while other one is excluded. It is our advantage to generate a good segmentation result for each frame merely based on the candidates calculated in Section 2.1. However, spectral clustering used to provide a foreground object model in Lee's method [16] might result in a bad result. As the authors write in the paper [16], the cluster ranked the fourth provides the best result, but not the first. Whereas there are still several misclassifications in the best cluster. To quantitatively analyze the performance of our method, we calculate the segmentation error of each video compared with the ground truth, and this enables a statistical evaluation of our method.

We compare the results with three state-of-the-art unsupervised methods [21,20,16], and two supervised methods [29,31] which require the first frame to be annotated. The results in Table 1 are the segmentation error as measured by the average number of incorrect pixels per frame compared to the ground-truth. Lower value means that the segmentation result is better. The segmentation results of our method are better than Ma [20] and Lee [16]. The main reason is that [20] and [16] cannot find primary object proposals in all frames. By contrast, the result of Zhang [21] is better. Taking cheetah as an example, segmentation is based on the original proposals and using object proposals generated by [17] sometimes is not enough. Zhang et al. [21] propose a new method called object proposal generation which is used to initialize the video object segmentation process, and augments this step to the framework. This step is additional and optimizes the original proposals. Through this process, the average per frame segmentation error of [21] is lowest compared with other methods among several videos. Moreover, with the exception of birdfall, the pixel error is larger probably because the foreground object color is close to background and the region producted by [17] is not accurate.

We perform a short analysis of the computational efficiency for each method. As the first step, all methods adopt the same algorithm [17] to find object-like regions in each video. After that, Lee [16] adopts spectral clustering method to find the cluster with the highest mean score, which is used for video object segmentation. Meanwhile, Ma [20] models the video segmentation as finding constrained Maximum Weight Cliques problem. It uses $K \times N$ nodes to construct a large graph, and therefore has high computational complexity. Zhang [21] builds binary edges from an ending node to the next three subsequent frames in Layered DAG. Compared with Zhang [21], our method just builds edges from one node to the next frame. The computational complexity for dynamic programming algorithm [21] is $O(n+m)$, in which n is the number of nodes and m is the number of edges. The shortest path algorithm that our method used can also solve by dynamic programming in linear time. Zhang [21] and our method have same computational complexity, but our method has less running time due to the less number of edges.

Table 1
Segmentation error on SegTrack dataset. We compare our method (Ours) with three state-of-the-art unsupervised methods ([21,20] and [16]), and two supervised methods ([29] and [31]) which require the first frame to be annotated.

Video	Ours	[21]	[20]	[16]	[29]	[31]
birdfall	267	155	189	288	252	454
cheetah	799	633	806	905	1142	1217
girl	1582	1488	1698	1785	1304	1755
monkeydog	398	365	472	521	563	683
parachute	197	220	221	201	235	502
Avg.	508	452	542	592	594	791

3.3. Qualitative analysis

In this subsection, we give visual comparison results between our framework and other competitors. We firstly show the results on the Berkeley Motion Segmentation Dataset.

In this paper, we test many videos in each group and show eight video segmentation results of this dataset. It is observed that our method generates a good foreground object for each frame. Such as video Marple1, either the optimal or suboptimal proposal is chosen as the foreground object. Other details about the experimental results are shown in Fig. 5. We compute precision

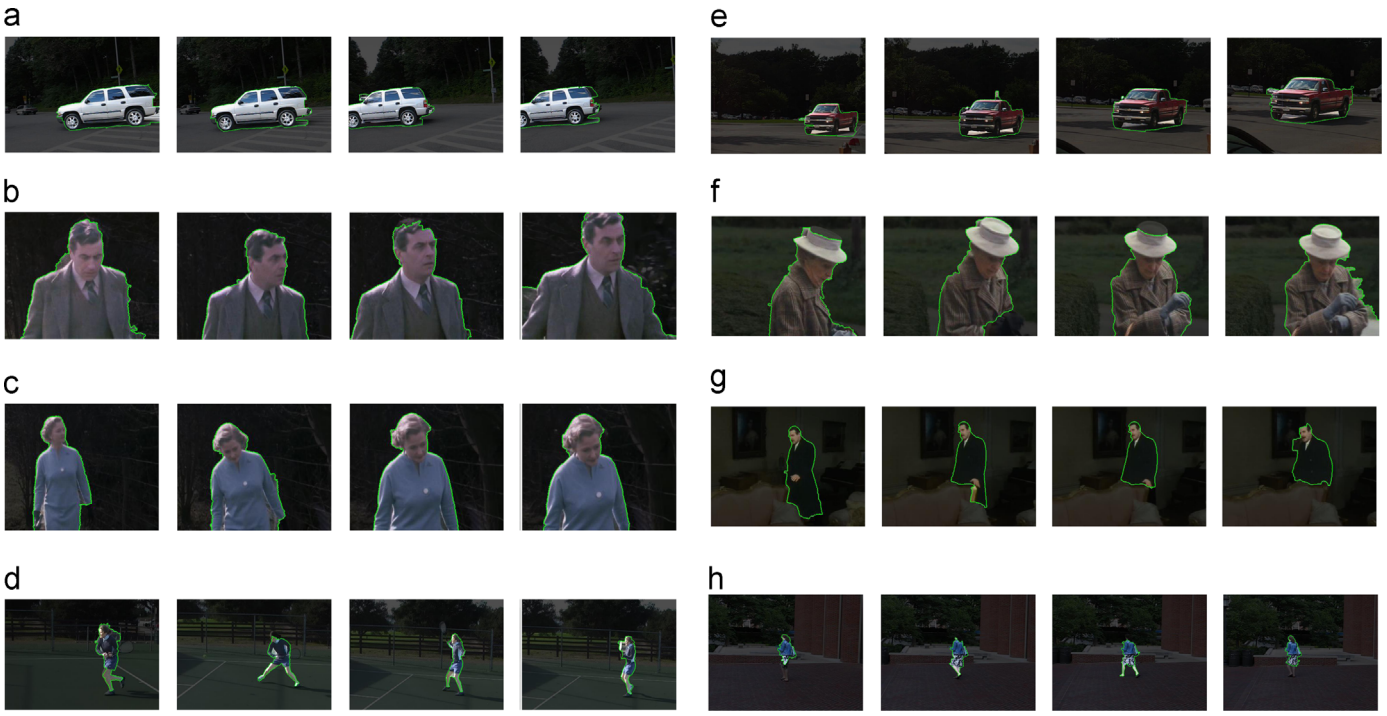


Fig. 5. BMSD segmentation results. Segmented primary foreground objects are marked by green boundaries. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.) (a) cars1 (b) marple1 (c) marple3 (d) tennis (e) cars8 (f) marple7 (g) marple11 (h) people1.

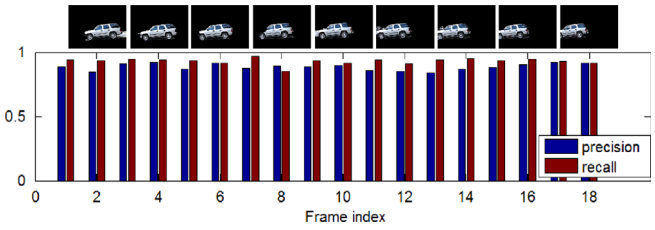


Fig. 6. Precision and recall for each frame in example Cars1 in BMSD. Besides, our segmentation results for each even frame are shown above the histogram.

Table 2
Precision and recall for the videos in BMSD.

Name	Precision	Recall
cars1	0.8855	0.9291
marple1	0.8990	0.9191
people1	0.9676	0.5333
tennis	0.9701	0.4139

and recall of the first video in each group. Take Car1 as an example, the histogram of precision and recall for each frame are shown in Fig. 6. The corresponding segmentation results to the even frames are shown above. It is observed that our method generates a good foreground object for each frame as the precision and recall rates remain stable around 0.9. Other details about the experimental results are shown in Table 2. It is observed that our method produces a relatively good result on this dataset.

Finally, we evaluate the proposed approach on GaTech video segmentation dataset. Fig. 7 shows the results of method [29] and our method. We can observe that the method [29] has a good result on many frames. But this method is supervised and does not use object-level segmentation, which leads to over-segmentation. Obviously, our results could segment the true foreground object from the background and outperform the state-of-the-art.

Beyond the database mentioned above, we also test some videos recorded by us, such as Fig. 1. The first two rows are the input video frames and our segmentation results. The result by Lee method [16] is shown in the last row. As we expected, the foreground object cannot be guaranteed to exist in all the frames because of the spectral clustering they use. It is observed that in frame 1, the running guy is cut out, for he has a bigger motion score. We use yellow arrow to label finding the right frame, but wrong segmentation. For frame No. 5, Lees method fails to provide a result. Here, we choose the closet frame (frame No. 4), which is marked by the red arrow to indicate the wrong frame and wrong segmentation. For frames 9 and 13, there are even no segmentation results. In contrast, our method works well under such scene with big motion interferent object.

4. Conclusion

In this paper, we present an unsupervised method of video object segmentation. The method we proposed is unsupervised which can automatically segment the foreground object in videos without any prior knowledge. The algorithm is based on object-level, while existing methods are normally based on low level information. Firstly, we generate object-like candidates in the every frame of unannotated video using method [17]. Then based on the corresponding map between the successive frames, the video segmentation problem is converted to graph model. After that, we use the shortest path method and the iterative approach with global model to obtain the foreground object proposals. Finally, a more precise pixel-level foreground object segmentation is performed.

The proposed method has a good performance compared with the state-of-the-art. In the future, we desire to apply our method to some vision applications, such as video retrieval.

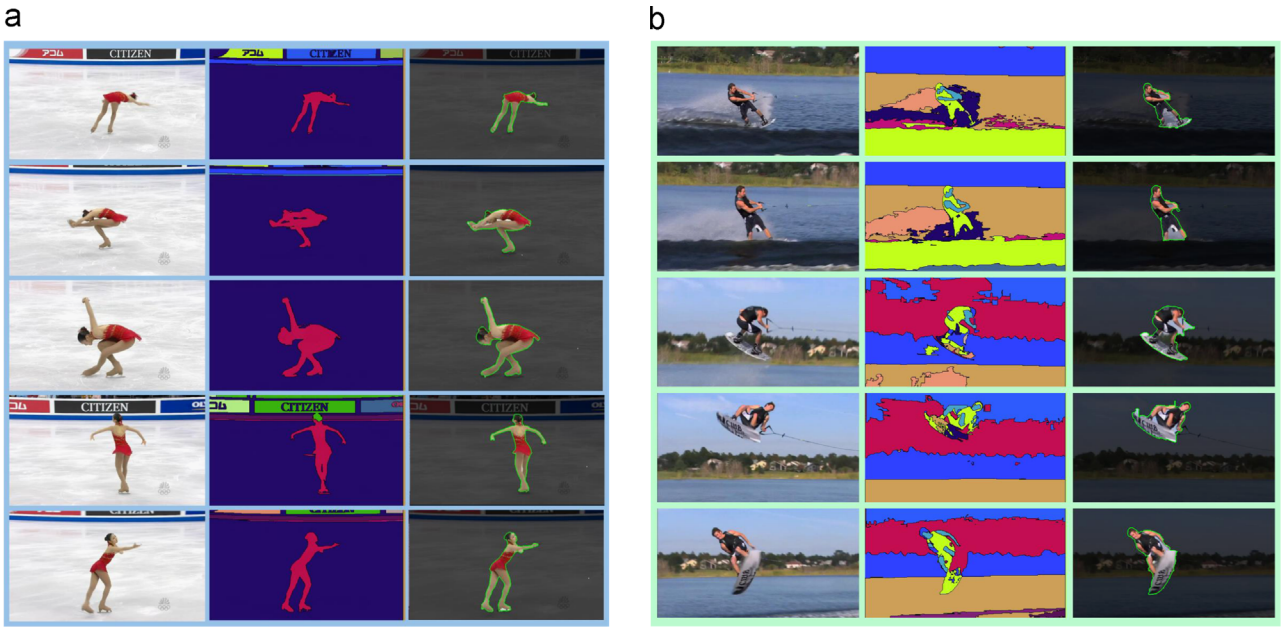


Fig. 7. GaTech segmentation dataset results. The first column is input frames. Our result is shown in the last column, compared with Tsai [29] result shown in the second column. Segmented primary foreground objects are marked by green boundaries. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.) (a) yunakim (b) waterski

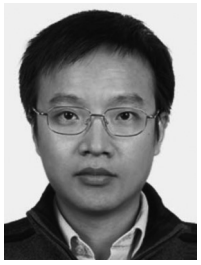
Acknowledgements

This work was supported by National Natural Science Foundation of China (No.61422213,61332012), National Basic Research Program of China (2013CB329305), National High-tech R&D Program of China (2014BAK11B03), and 100 Talents Programme of the Chinese Academy of Sciences.

References

[1] P. Ochs, J. Malik, T. Brox, Segmentation of moving objects by long term video analysis, IEEE Trans. Pattern Anal. Mach. Intell. 36 (2013) 1187–1200.
[2] Y.J. Lee, J. Ghosh, K. Grauman, Discovering important people and objects for egocentric video summarization., in: CVPR, vol. 1, 2012, pp. 1346–1353.
[3] D. Potapov, M. Douze, Z. Harchaoui, C. Schmid, et al., Category-specific video summarization, in: ECCV, 2014.
[4] Nataraj Jammalamadaka, Andrew Zisserman, Marcin Eichner, Vittorio Ferrari, and C. V. Jawahar. 2012. Video retrieval by mimicking poses. In ICMR, ACM, New York, NY, USA.
[5] C.C. Loy, T. Xiang, S. Gong, Time-delayed correlation analysis for multi-camera activity understanding, Int. J. Comput. Vis. 90 (2010) 106–129.
[6] B.T. Morris, M.M. Trivedi, Trajectory learning for activity understanding: unsupervised, multilevel, and long-term adaptive approach, IEEE Trans. Pattern Anal. Mach. Intell. 33 (2011) 2287–2301.
[7] X. Cui, Q. Liu, D. Metaxas, Temporal spectral residual: fast motion saliency detection, in: Multimedia, 2009, pp. 617–620.
[8] C. Guo, Q. Ma, L. Zhang, Spatio-temporal saliency detection using phase spectrum of quaternion Fourier transform, in: CVPR, 2008, pp. 1–8.
[9] E. Rahtu, J. Kannala, M. Salo, J. Heikkilä, Segmenting salient objects from images and videos, in: ECCV, 2010, pp. 366–379.
[10] H. Fu, X. Cao, Z. Tu, Cluster-based co-saliency detection, IEEE Trans. Image Process. 22 (2013) 3766–3778.
[11] Z. Gu, T. Mei, X.-S. Hua, X. Wu, S. Li, Ems: energy minimization based video scene segmentation, in: ICME, 2007, pp. 520–523.
[12] J. Wang, X. Tian, L. Yang, Z.-J. Zha, X.-S. Hua, Optimized video scene segmentation, in: 2008 IEEE International Conference on Multimedia and Expo, 2008, pp. 301–304.
[13] T. Brox, J. Malik, Object segmentation by long term analysis of point trajectories, in: ECCV, 2010, pp. 282–295.
[14] P. Ochs, T. Brox, Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions, in: ICCV, 2011, pp. 1583–1590.
[15] W. Brendel, S. Todorovic, Video object segmentation by tracking regions, in: ICCV, 2009, pp. 833–840.
[16] Y. J. Lee, J. Kim, K. Grauman, Key-segments for video object segmentation, in: ICCV, 2011, pp. 1995–2002.
[17] I. Endres, D. Hoiem, Category independent object proposals, in: ECCV, 2010, pp. 575–588.
[18] J. Carreira, C. Sminchisescu, Constrained parametric min-cuts for automatic object segmentation, in: CVPR, 2010, pp. 3241–3248.
[19] B. Alexe, T. Deselaers, V. Ferrari, What is an object?, in: CVPR, 2010, pp. 73–80.
[20] T. Ma, L.J. Latecki, Maximum weight cliques with mutex constraints for video object segmentation, in: CVPR, 2012, pp. 670–677.
[21] D. Zhang, O. Javed, M. Shah, Video object segmentation through spatially accurate and temporally dense extraction of primary object regions, in: CVPR, 2013.
[22] V. Kolmogorov, Convergent tree-reweighted message passing for energy minimization, IEEE Trans. Pattern Anal. Mach. Intell. 28 (2006) 1568–1583.
[23] B. Zhang, H. Zhao, X. Cao, Video object segmentation with shortest path, in: Multimedia, 2012, pp. 801–804.
[24] C. Liu, et al., Beyond pixels: exploring new representations and applications for motion analysis (Ph.D. thesis), Massachusetts Institute of Technology, 2009.
[25] C. Rother, V. Kolmogorov, A. Blake, Grabcut: interactive foreground extraction using iterated graph cuts, ACM Trans. Graph. 23 (2004) 309–314.
[26] Y.Y. Boykov, M.-P. Jolly, Interactive graph cuts for optimal boundary & region segmentation of objects in nd images, in: ICCV, vol. 1, 2001, pp. 105–112.
[27] J. Kim, K. Grauman, Boundary preserving dense local regions, in: CVPR, 2011, pp. 1553–1560.
[28] Y. Boykov, O. Veksler, R. Zabih, Fast approximate energy minimization via graph cuts, IEEE Trans. Pattern Anal. Mach. Intell. 23 (2001) 1222–1239.
[29] D. Tsai, M. Flagg, J.M. Rehg, Motion coherent tracking with multi-label mrf optimization, in: BMVC, 2010.
[30] M. Grundmann, V. Kwatra, M. Han, I. Essa, Efficient hierarchical graph-based video segmentation, in: CVPR, 2010, pp. 2141–2148.
[31] P. Chockalingam, N. Pradeep, S. Birchfield, Adaptive fragments-based tracking of non-rigid objects using level sets, in: ICCV, 2009, pp. 1530–1537.
[32] H. Fu, D. Xu, B. Zhang, and S. Lin, "Object-Based Multiple Foreground Video Co-segmentation," in CVPR, 2014, pp. 3166–3173.
[33] H. Fu, D. Xu, S. Lin, and J. Liu, "Object-based RGBD Image Co-segmentation with Mutex Constraint," in CVPR, 2015, Boston, US.

Xiaochun Cao is a professor at the Institute of Information Engineering, Chinese Academy of Sciences since 2012. He received the B.E. and M.E. degrees both in computer science from Beihang University(BUAA), China, and the Ph.D. degree in computer science from the University of Central Florida, USA, with his dissertation nominated for the university-level Outstanding Dissertation Award. After graduation, he spent about three years at ObjectVideo Inc. as a Research Scientist. From 2008 to 2012, he was a professor at Tianjin University. He has authored and coauthored over 50 journal and conference papers. In 2004 and 2010, Dr. Cao was the recipients of the Piero Zamperoni best student paper award at the International Conference on Pattern Recognition.



Feng Wang received the B.E. and M.E. degrees both in School of Computer Software, Tianjin University, TianJin, China, in 2012 and 2015. Her research interests include computer vision, such as video segmentation and image retrieval.

Bao Zhang received the B.E. degree in software engineering in the School of Computer Software,Tianjin University, Tianjin, China, in 2010, and the M. E. degree with School of ComputerScience and Technology, Tianjin University, in 2013. His current research interestsinclude computer vision, scene classification, video processing, image saliency detection and segmentation.

Huazhu Fu is a Research Fellow in School of Computer Engineering at Nanyang Technological University (NTU), Singapore.He received the B.S. degree from Nankai University in 2006, the M.E. degree from Tianjin University of Technology in 2010, and the Ph.D. degree from Tianjin University, China, in 2013. His current research interests include computer vision, medical image processing, image saliency detection and segmentation.