Fast Appearance Modeling for Automatic Primary Video Object Segmentation

Jiong Yang, Brian Price, *Member, IEEE*, Xiaohui Shen, *Member, IEEE*, Zhe Lin, *Member, IEEE*, and Junsong Yuan, *Senior Member, IEEE*

***Abstract*— Automatic segmentation of the primary object in a video clip is a challenging problem as there is no prior knowledge of the primary object. Most existing techniques thus adapt an iterative approach for foreground and background appearance modeling, i.e., fix the appearance model while optimizing the segmentation and fix the segmentation while optimizing the appearance model. However, these approaches may rely on good initialization and can be easily trapped in local optimal. In addition, they are usually time consuming for analyzing videos. To address these limitations, we propose a novel and efficient appearance modeling technique for automatic primary video object segmentation in the Markov random field (MRF) framework. It embeds the appearance constraint as auxiliary nodes and edges in the MRF structure, and can optimize both the segmentation and appearance model parameters simultaneously in one graph cut. The extensive experimental evaluations validate the superiority of the proposed approach over the state-of-the-art methods, in both efficiency and effectiveness.**

***Index Terms*— Automatic, primary, video, object, segmentation, graph cut, appearance modeling.**

I. Introduction

THE PRIMARY object in a video sequence can be defined as the object that is locally salient and present in most of the frames [42], [44]. The target of automatic primary video object segmentation is to segment out the primary object in a video sequence without any human intervention. It has a wide range of applications including video object recognition, action recognition and video summarization. Some examples are shown in Fig. 1. The existing works on video object segmentation can be divided into two groups based on the amount of human intervention required: interactive segmentation [4], [15] and fully automatic segmentation [18], [20], [29], [44]. Our method belongs to the latter and does not assume the object is present in all the frames.

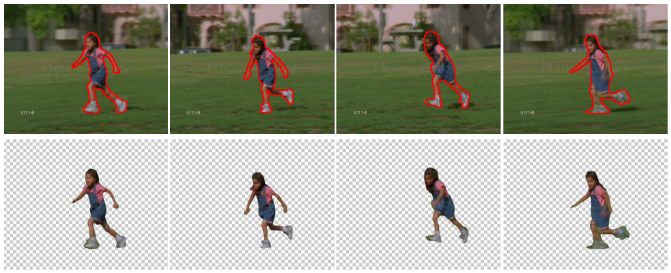


Fig. 1. Illustration of primary object segmentation in videos. The top row is the original video frames with the expected segmentation results rendered as red contours. The bottom row is the same segmentation results after removing the background.

Following the outstanding performance of Markov Random Field (MRF) based methods in image object segmentation [9], [32], [35], many of the existing video object segmentation approaches also build spatio-temporal MRF graphs and show promising results [15], [29], [44]. These approaches build a spatio-temporal graph by connecting spatially or temporally connected regions, *e.g.,* pixels [35] or superpixels [29], and cast the segmentation problem into a node labeling problem in a Markov Random Field. This process is illustrated graphically in Fig. 2. Such automatic primary video object segmentation methods usually have three major steps: initial visual or motion saliency estimation, spatio-temporal graph connection and foreground/background appearance modeling. Automatic foreground/background appearance modeling is important as the saliency estimation is usually noisy especially along object boundaries due to cluttered background or background motions. However, it is challenging because there is no prior knowledge about foreground and background regions. Formally, with the presence of appearance constraints, there are two groups of parameters in the optimization process, *i.e.,* segmentation labels $\mathbf{x}$ and appearance model Θ. For many commonly used appearance models such as Gaussian Mixture Models (GMM) [29] or Multiple Instance Learning [39], it is intractable to solve both parameters simultaneously. Hence, many existing methods adapt an iterative approach. They use the segmentation result of the previous iteration to train foreground and background appearance models which are then used to refine the segmentation in the next iteration. However, these methods can be easily trapped in local optimal and are time consuming especially for video data.

Manuscript received March 26, 2015; revised July 15, 2015 and September 21, 2015; accepted October 28, 2015. Date of publication November 13, 2015; date of current version December 23, 2015. This work was supported in part by the Adobe Gift Grant, and in part by the Singapore Ministry of Education Academic Research Fund Tier 1 under Grant M4011272.040. This research was carried out in the Rapid-Rich Object Search Laboratory supported by the National Research Foundation, Prime Ministers Office, Singapore, under its IDM Futures Funding Initiative and administered by the Interactive and Digital Media Programme Office. The associate editor coordinating the review of this manuscript and approving it for publication was Mr. Pierre-Marc Jodoin.

J. Yang is with the Rapid Rich Object Search Laboratory, Nanyang Technological University, Singapore 637553 (e-mail: yang0374@e.ntu.edu.sg).

B. Price, X. Shen, and Z. Lin are with Adobe Research, San Jose, CA 95110 USA (e-mail: bprice@adobe.com; xshen@adobe.com; zlin@adobe.com).

J. Yuan is with the Department of Electrical and Electronics Engineering, Nanyang Technological University, Singapore 639798 (e-mail: jsyuan@ntu.edu.sg).

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/TIP.2015.2500820

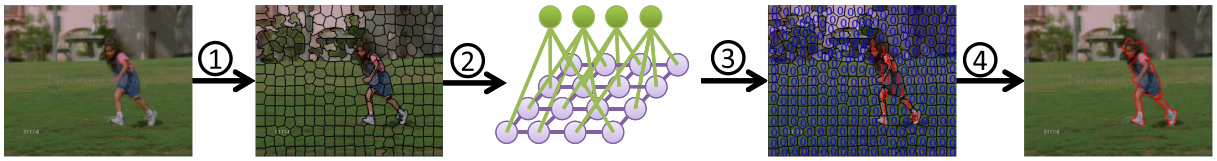


Fig. 2. The overall work flow of the proposed segmentation framework. 1. Superpixel segmentation; 2. Graph construction: the purple nodes and edges represent the superpixels and the spatio-temporal neighborhood connections between them. They are used to encourage the spatio-temporal smoothness of the segmentation. The green nodes and edges represent the auxiliary nodes and connections for appearance modeling. They are used to encourage the appearance coherence and disparity within and between the foreground and background regions, respectively; 3: Node labeling by MRF inference; 4: Final segmentation result.

Recently, [35] proposed an appearance modeling technique in the graph based interactive image segmentation framework which can solve both the segmentation labels and appearance model parameters simultaneously without iteration. In their approach, they model each pixel as a node and quantize it into a bin in the RGB histogram space. It shows that when the foreground and background appearance are represented non-parametrically in the RGB histogram space, the appearance constraint is equivalent to adding auxiliary nodes and edges to the original MRF structure. However, due to the fundamental difference between image data and video data, the original approach in [35] is not practically applicable to video because it requires each node to be described by a single bin in the histogram space. For video object segmentation, superpixels are generally used due to the large data volume and more robust features like SIFT [23] or Textons are beneficial to better capture the viewpoint and lighting variations between different frames. As a result, each pixel will now have multiple features and each node will correspond to multiple pixels. Hence, in this paper, we extend the efficient appearance modeling technique in [35] to primary video object segmentation by addressing these challenges. The proposed appearance modeling technique is more general than [35] and can handle all the above mentioned difficulties. The resultant auxiliary connections are also different from [35] because in [35] each pixel node is connected to one auxiliary node, while in our approach each superpixel node can be connected to multiple auxiliary nodes. Experimental evaluations validate the superiority of the proposed approach over directly applying [35] for automatic primary video object segmentation.

In summary, the major contribution of this paper is that we propose an efficient and effective appearance modeling technique in the MRF based segmentation framework for primary video object segmentation. It embeds the appearance constraint directly into the graph by adding auxiliary nodes/connections, and the resultant graph-partition problem can be solved efficiently by one graph cut. Although inspired by the idea of [35], we have made the non-trivial extension from static images to videos, and we generalize the framework in more complicated cases.

In the following sections of this paper, we will first discuss the related works in Section II. Then we will present, in Section III, the entire graph structure for primary video object segmentation and emphasize how we formulate and optimize both the label and appearance model parameters simultaneously. The proposed method is evaluated in Section IV on two benchmark datasets and compared with the recent state of the art. The entire paper is concluded at Section V.

II. Related Work

A. Low Level Video Segmentation

Common low level video segmentation methods include superpixel segmentation [1], [38] and supervoxel segmentation [14], [40]. Superpixel segmentation methods typically over-segment the entire frame into visually coherent groups or segments. Supervoxel segmentation is similar to superpixel segmentation but also groups pixels temporally and, hence, produces spatio-temporal segments. Note that in this paper we are primarily interested in object level segmentation instead of unsupervised low level pixel grouping. Actually, superpixels and supervoxels are usually used as the primitive input in place of pixels in the context of video object segmentation for efficiency [15], [29], [39]. Another type of low level segmentation is object proposal segmentation [8], [11], [31]. It produces a large set of candidate segments that are likely to contain semantic objects. However, they aim at a high recall instead of precision and are generally computationally expensive compared with the superpixel or supervoxel methods. Many high level video object segmentation methods use these proposals as the primitive input [12], [18], [19], [26], [44], [45].

B. Object Level Video Segmentation

The existing works related to video object segmentation, *e.g.,* [12], [15], [28], [29], can be divided into 3 groups, *i.e.,* interactive video object segmentation, automatic video object segmentation and video object co-segmentation.

As briefly described in the introduction section, interactive video object segmentation requires human intervention in the segmentation process. Some of these approaches require the user to provide a pixel-wise segmentation on the first few frames for initialization [3], [15], [30], [37], while others require the user to continuously correct the segmentation errors [4], [21]. These methods generally require a considerable amount of human effort and, hence, are not scalable to large video collections.

In contrast, automatic video object segmentation does not require any human intervention and tries to automatically infer where the primary object is from the various cues including saliency, spatio-temporal smoothness and foreground/background appearance coherency [5], [18], [19], [26], [27], [29], [44]. The most related approach is [22] as it also relies on saliency estimation and builds spatio-temporal graph by connecting neighborhood superpixels. However, it uses

color GMMs to model the local foreground and background appearance separately in an iterative manner. Several papers [5], [18], [19], [26], [44] use object proposals [11] as the primitive input which contribute significantly to the inefficiency of these methods. The method in [18] first uses spectral clustering to group proposals with coherent appearance and then train foreground/background color GMMs and object location priors. Pixel-wise graph cut is used to produce the final segmentation mask for each individual frame. Ma and Latecki [26] adapt a similar pipeline with [18] but use constrained maximum weighted cliques to group proposals. The method in [44] builds a spatial-temporal graph by connecting proposals and uses dynamic programming to find the most confident trajectory. It then uses pixel-wise graph cut to refine the final segmentation mask for each individual frame based on the found proposal trajectory. The method in [5] produces multiple proposal chains by linking local segments using long-range temporal constraints. It then obtains the final segmentation result by pixel-wise per-frame MRF smoothing using the appearance and location priors learned from these initial chains. The method in [19] tracks the proposals temporally using incremental regression and refines the final segmentations by composite statistic inferences. The method in [27] explores this problem in MPEG2 compressed domain. On the P-Frames, it computes the motion saliency priors by compensating camera motion. On the I-Frames, it computes the color-based segmentation by morphological approach. These two cues are then merged and followed by a spatio-temporal filtering using quadric surfaces to give the final segmentation result. The method in [25] first segments the selected key frames into an over complete set of segments using image segmentation algorithms like [33] and then employs the cohesive sub-graph mining technique to find the salient segments with similar appearance and strong mutual affinity. Zhao *et al.* [46], [47] adapt a similar pipeline but use topic model to discover the coherent segments. Both methods disregard the temporal smoothness of the object region and only aim at the rough location instead of accurate segmentation.

Video object co-segmentation is also automatic but tries to seek supervision by assuming the primary object is present in a batch of given videos [12], [39], [45]. Both [12] and [39] formulate the segmentation as node selection or labeling in spatio-temporal graph, while [45] finds the maximum weighted clique in a completely connected graph. The method in [12] does not have an explicit global appearance model, and [39] adapts the iterative appearance modeling approach using multiple instance learning.

C. Appearance Models in MRF Segmentation Framework

In the existing image or video object segmentation frameworks using MRF structure, the most commonly used appearance model is color GMM which models the foreground and background appearances separately [5], [15], [18], [26], [29], [32], [44]. Multiple instance learning on context features is also used in [39] to model the foreground and background appearance in a discriminative manner. However, all the aforementioned works adapt an iterative approach to gradually refine the appearance model and segmentation labels. Recently, [35] proposed to use color histograms to model the appearance non-parametrically for static image segmentation. Both the appearance model and segmentation labels can be optimized simultaneously without iteration.

III. Proposed Approach

In this section, we introduce the proposed approach for automatic primary video object segmentation. The input is a plain video clip without any annotations and the output is a pixel-wise spatio-temporal foreground *v.s.* background segmentation of the entire sequence. Similar to many existing image and video object segmentation approaches, we cast the segmentation to a two-class node labeling problem in a Markov Random Field. Within the MRF graph, each node is modeled as a superpixel, and will be labeled as either foreground or background in the segmentation process. The overall work flow is shown in Fig. 2. In this work, we first segment each video frame into a set of superpixels using the SLIC algorithm [1] and then represent each node in the MRF as a superpixel. We typically have around 2500 superpixels per video frame. We choose not to use pixels because the computational and memory cost will be high for video data in our framework. Meanwhile superpixels produced by SLIC [1] can preserve most of the boundaries, and over-segmentation is not a critical concern.

In the following, we use s_i^j to denote the j^{th} superpixel of the i^{th} frame, N to denote the total number of frames and M_i to denote the number of superpixels in the i^{th} frame. The segmentation target is to assign each superpixel s_i^j a label x_i^j indicating if it is foreground, $x_i^j = 1$, or background, $x_i^j = 0$. The overall optimization formulation in terms of the graph energy minimization is expressed as

$$\mathbf{x}^* = \arg\min_{\mathbf{x},\Theta} E(\mathbf{s}, \mathbf{x}, \Theta) \tag{1}$$

where $E(\mathbf{s}, \mathbf{x}, \Theta)$ is defined as

$$E(\mathbf{s}, \mathbf{x}, \Theta) = \Phi_u(\mathbf{s}, \mathbf{x}) + \alpha_p \times \Phi_p(\mathbf{s}, \mathbf{x}) + \alpha_a \times \Phi_a(\mathbf{s}, \mathbf{x}, \Theta). \tag{2}$$

The vector $\mathbf{x}$ and Θ denote the $\{0, 1\}$ labeling of all the superpixels and the appearance model parameters, respectively, $\mathbf{s}$ denotes the collection of all the superpixels and Φ_u, Φ_p and Φ_a denote the unary potential, pairwise potential and appearance constraint potential, respectively. α_p and α_a are two weight parameters for linear combination.

A. Unary Potentials

Since saliency has been proven to be effective in highlighting the primary object in a completely automatic setting by simulating where human looks [2], [16], [24], [27], [29], [43], we use it to model the unary potential of each node. In order to capture different aspects of saliency, four saliency estimations are employed including both appearance and motion saliency, *i.e.,* AMC image saliency [34], GBMR image saliency, [41], GC motion saliency [42] and W motion saliency [42]. To produce a single saliency estimation for each frame, we combine these saliency maps by weighted

linear combination where the weight of each saliency map is determined by the SVM-Fusion technique proposed in [42]. The SVM-Fusion technique can adaptively predict the quality of each saliency map without using ground truth, and thus the weighted combination can adaptively reject noise and emphasize the most proper saliency cues for each individual frame. We also warp the saliency estimations along the optical flow direction to encourage temporal smoothness. The saliency value of a superpixel is then computed as the average saliency value of the contained pixels. An alternative is to use the peak saliency value instead of the average. However, we did not find these two approaches are statistically different under the paired t-test with a significance level of 0.05. Let $A(s_i^j)$ denote the saliency value of superpixel s_i^j, its unary potential is given by:

$$\phi_u(s_i^j) = \begin{cases} -\log(A(s_i^j)) & \text{if } x_i^j = 1 \\ -\log(1 - A(s_i^j)) & \text{if } x_i^j = 0. \end{cases} \tag{3}$$

The total unary term in Eq.(2) can be computed as

$$\Phi_u(\mathbf{s}, \mathbf{x}) = \sum_i^N \sum_j^{M_i} \phi_u(s_i^j). \tag{4}$$

This definition implies that it is costly to label a highly salient superpixel as background and vice versa.

B. Pairwise Potentials

There are two types of neighborhood relationships between superpixels in videos, *i.e.,* spatial neighborhoods and temporal neighborhoods. Two superpixels are spatially connected if they share a common edge and temporally connected if they have pixels linked by optical flow. In the MRF graph, only neighboring superpixels will have nonzero edge and the edge weight represents the cost induced by assigning different labels to the connected superpixels. Hence, the edge weight is usually measured as the inverse likelihood of the existence of a real edge between two superpixels. Apart from using local similarity, we also use the high level edge detection in both the appearance and motion domain to determine the edge weight. More specifically, we use color and optical flow orientation histogram to compute the local similarity and the structural forest edge detector [10] to compute the edge strengths. Note that, to detect motion boundaries for each frame, we first convert the XY dense flow vector of each pixel to a color representation using the method proposed in [22] and then apply the edge detection in the color domain. The appearance and motion edge maps are then combined by the maximum operation. Overall, the spatial and temporal pairwise potentials between neighboring superpixels are computed as

$$\begin{aligned} \phi_s(s_i^j, s_p^q) &= (1 - e(s_i^j, s_p^q)) \times (1 - \delta(x_i^j, x_p^q)) \\ &\quad \times \exp(-\beta_s^{-1} \|\mathbf{F_i^j} - \mathbf{F_p^q}\|^2) \\ \phi_t(s_i^j, s_p^q) &= c(s_i^j, s_p^q) \times (1 - \delta(x_i^j, x_p^q)) \\ &\quad \times \exp(-\beta_t^{-1} \|\mathbf{H_i^j} - \mathbf{H_p^q}\|^2). \end{aligned} \tag{5}$$

Here, $e(s_i^j, s_p^q)$ denotes the average edge strength between superpixel s_i^j and s_p^q, $c(s_i^j, s_p^q)$ denotes the percentage of pixels in s_p^q that are linked to s_i^j by optical flow, and δ is the standard Kronecker delta function, *i.e.,* $\delta(u, v) = 1$ if $u = v$ and $\delta(u, v) = 0$ if $u \neq v$. $\mathbf{F_i^j}$ is the concatenation of color and optical flow orientation histogram and $\mathbf{H_i^j}$ is the color histogram. The motion feature is only included in the spatial pairwise potentials because temporal pairs correspond to superpixels in different frames. The overall pairwise potential is then computed as the weighted summation of all the spatial and temporal pairwise terms:

$$\begin{aligned} \Phi_p(\mathbf{s}, \mathbf{x}) = \alpha_s \times \sum_{\{s_i^j, s_p^q\} \in \mathcal{N}_s} \phi_s(s_i^j, s_p^q) \\ + \alpha_t \times \sum_{\{s_i^j, s_p^q\} \in \mathcal{N}_t} \phi_t(s_i^j, s_p^q) \end{aligned} \tag{6}$$

where $\mathcal{N}_s$ and $\mathcal{N}_t$ denote the collections of all the spatial and temporal neighborhood pairs, respectively. α_s and α_t are two weight parameters for linear combination.

C. Appearance Auxiliary Potential

In general, the appearance constraint $\Phi_a(\mathbf{s}, \mathbf{x}, \Theta)$ in Eq.(2) can be written as $\Phi_a(\mathbf{s}, \mathbf{x}, \Theta) = f(\mathbf{s}, \mathbf{x}, g(\mathbf{s}, \mathbf{x}))$ where f measures how consistent the current labeling $\mathbf{x}$ is with the appearance model, and g computes the appearance model parameters given the current labeling $\mathbf{x}$. However it is impossible to have an analytical expression to $\Phi_a(\mathbf{s}, \mathbf{x}, \Theta)$ for many popular appearance models because the appearance model training usually involves complicate optimization process, *e.g.,* EM optimization in GMM, so for such methods Eq.(1) cannot be solved analytically. Hence, an alternative optimization scheme is usually employed to solve Eq.(1), *i.e.,* fix the appearance model while solving $\mathbf{x}$ and fix $\mathbf{x}$ while optimizing the appearance model. Inspired by [35], in this work we propose an appearance model for video object segmentation in which $\Phi_a(\mathbf{s}, \mathbf{x}, \Theta)$ can be expressed analytically in terms of $\mathbf{x}$, and Eq.(1) can be solved efficiently by one graph cut. In the following, we first review the method of [35] on static image segmentation and then discuss the challenges in adapting the idea to videos and how we overcome them.

The method in [35] models each pixel as a node and represents each node as a single bin in the RGB histogram space for appearance modeling. Let p_i and x_i denote the i^{th} pixel and its label, respectively, b_i denote the assigned bin of pixel p_i, H denote the dimensionality of the histogram space and P denote the total number of pixels. Furthermore we use $\mathbf{\Omega}_F^k$ and $\mathbf{\Omega}_B^k$ to denote the number of pixels assigned to the k^{th} bin in the foreground and background regions, respectively, and $\mathbf{\Omega}^k$ to denote the number of pixels assigned to the k^{th} bin in the entire image, *i.e.,* $\mathbf{\Omega}_F^k = |\{p_i | x_i = 1\}|$, $\mathbf{\Omega}_B^k = |\{p_i | x_i = 0\}|$ and $\mathbf{\Omega}^k = \mathbf{\Omega}_F^k + \mathbf{\Omega}_B^k$ where $|.|$ denotes the Cardinality of a set. Then the foreground and background probability of the k^{th} histogram bin is given by $p(F|k) = \frac{\mathbf{\Omega}_F^k}{\mathbf{\Omega}^k}$ and $p(B|k) = \frac{\mathbf{\Omega}_B^k}{\mathbf{\Omega}^k}$, respectively. Finally, the appearance constraint potential of each pixel p_i can be computed as

$$\phi_a(p_i) = \begin{cases} -\ln p(F|b_i) & \text{if } x_i = 1 \\ -\ln p(B|b_i) & \text{if } x_i = 0. \end{cases} \tag{7}$$

Then the total appearance constraint potential of all the pixels, *i.e.,* the last term in Eq.(2), can be computed as

$$\begin{aligned}\Phi_a &= \sum_{i=1}^{P} \phi_a(p_i) \\ &= \sum_{i=1}^{P} -\delta(x_i, 1) \times \ln p(F|b_i) - \delta(x_i, 0) \times \ln p(B|b_i) \\ &= -\sum_{i=1}^{P} (\delta(x_i, 1) \times \ln \frac{\Omega_F^{b_i}}{\Omega^{b_i}} + \delta(x_i, 0) \times \ln \frac{\Omega_B^{b_i}}{\Omega^{b_i}}) \\ &= -(\sum_{k=1}^{H} \Omega_F^k \times \ln \frac{\Omega_F^k}{\Omega^k} + \sum_{k=1}^{H} \Omega_B^k \times \ln \frac{\Omega_B^k}{\Omega^k}) \\ &= -\sum_{k=1}^{H} (\Omega_F^k \times \ln \frac{\Omega_F^k}{\Omega^k} + \Omega_B^k \times \ln \frac{\Omega_B^k}{\Omega^k}). \end{aligned} \tag{8}$$

The inner part of the summation in Eq.(8) can be approximated by $\left|\Omega_F^k - \Omega_B^k\right|$ since $\Omega_F^k + \Omega_B^k = \Omega^k$. Hence, $\Phi_a(\mathbf{x}, \Theta) \approx -\sum_{k=1}^{H} \left|\Omega_F^k - \Omega_B^k\right| = \sum_{k=1}^{H} 2\min(\Omega_F^k, \Omega_F^k) - \Omega^k$. As we are only interested in minimizing $\Phi_a(\mathbf{x}, \Theta)$ instead of its absolute value, we can drop the constant term Ω^k and the multiplier 2. Eventually, the appearance model is reduced to

$$\Phi_a(\mathbf{x}, \Theta) = \sum_{k=1}^{H} \min(\Omega_F^k, \Omega_B^k), \tag{9}$$

and the inner part of this summation is the number of pixels that are assigned to the k^{th} bin taking the minority label. Interestingly, this appearance term turns out to be equivalent to adding some auxiliary nodes and edges to the MRF graph. The addition procedure is simple: 1) add H auxiliary nodes in which each node corresponds to a bin of the histogram, and the unary potential of these newly added nodes are set to $-\log(0.5)$; 2) Connect each pixel to the auxiliary node that corresponds to its assigned bin. The rationality of this equivalence is that the auxiliary nodes are guaranteed to be labeled as the majority label of its connected pixels when the graph energy is minimized and, hence, the cost incurred by each auxiliary node is equal to the number of connected pixels taking the minority label.

A naive extension of [35] to our superpixel based video object segmentation is to take the mean RGB color of each superpixel and assign it to one of the bins in the color histogram space. However raw color features alone may not be robust enough to accurately capture the viewpoint and lighting variations between frames. Hence, we propose to use more advanced features, *i.e.,* SIFT and Texton, to measure the similarity between image regions. The fusion of these features has shown promising results in many vision problems such as [36], [42], and [48]. It is worth noting that these features are not computed only on the pixels within a superpixel, but rather are computed in a larger window around a pixel. For example, all SIFT features computed on all videos in our experiment cover more than a single superpixel. This means that important context information around the superpixels is contained in these features. However, for video object segmentation, both the original appearance modeling approach in [35] and its naive extension are not readily applicable to these multi-feature situations. This is because both SIFT and Texton are key point based features and are not confined to any arbitrarily shaped superpixel. Hence, we adapt a different approach to extract and fuse these features. We first extract these features around a set of key points defined by a dense grid, *e.g.,* sample a key point every 4 pixels horizontally and vertically. We then use the bag of words approach to quantize each type of feature to a particular bin and assign each key point to a single bin by taking the Cartesian Product of the different types of features. However, unlike the case of single pixels, each node will now contain more than a single feature point, and the original approach in [35] cannot handle this situation. Hence, we propose a variation of the original technique to handle the cases where each node is described by a set of bins, *i.e.,* a full histogram, instead of a single bin in the histogram. In the following, we will introduce this new method and prove that it can also be equated by adding auxiliary nodes and edges.

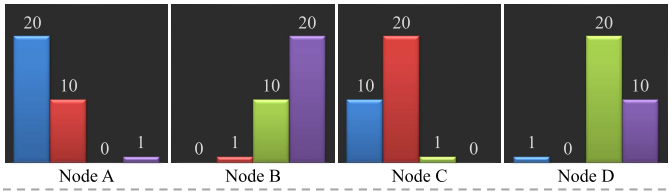
For consistency, we first redefine some of the terms used in the description of the pixel wise approach in [35]. Let $b_i^{j,k}$ denote the number of votes in the k^{th} bin of superpixel s_i^j's histogram, *i.e.,* the number of feature points in superpixel s_i^j that are assigned to the k^{th} bin, H denote the total number of bins in the histogram feature space, Ω_F^k and Ω_B^k denote the total number of votes in the k^{th} bin of the foreground and background superpixels, respectively and Ω^k denote the total number of votes in the k^{th} bin in all the superpixels, *i.e.,* $\Omega_F^k = \sum_i^N \sum_j^{M_i} \delta(x_i^j, 1) b_i^{j,k}$, $\Omega_B^k = \sum_i^N \sum_j^{M_i} \delta(x_i^j, 0) b_i^{j,k}$ and $\Omega^k = \Omega_F^k + \Omega_B^k$. We can then compute the foreground and background probability of the k^{th} bin as $p(F|k) = \frac{\Omega_F^k}{\Omega^k}$ and $p(B|k) = \frac{\Omega_B^k}{\Omega^k}$, respectively. With the Naive Bayes assumption on the feature points in a superpixel, we can compute the foreground and background probability of superpixel s_i^j as $p(F|s_i^j) = \prod_{k=1}^{H} p(F|k)^{b_i^{j,k}}$ and $p(B|s_i^j) = \prod_{k=1}^{H} p(B|k)^{b_i^{j,k}}$, respectively. Then the last term in Eq.(2) is computed as

$$\Phi_a(\mathbf{s}, \mathbf{x}, \Theta) = \sum_{i}^{N} \sum_{j}^{M_i} \phi_a(s_i^j), \tag{10}$$

where

$$\begin{aligned}\phi_a(s_i^j) &= \begin{cases} -\ln p(F|s_i^j) & \text{if } x_i^j = 1 \\ -\ln p(B|s_i^j) & \text{if } x_i^j = 0 \end{cases} \\ &= \begin{cases} -\sum_{k=1}^{H} b_i^{j,k} \times \ln p(F|k) & \text{if } x_i^j = 1 \\ -\sum_{k=1}^{H} b_i^{j,k} \times \ln p(B|k) & \text{if } x_i^j = 0. \end{cases} \end{aligned} \tag{11}$$

An example is shown in Fig. 3 to illustrate how this term can enforce the appearance constraints. From this figure, it can be seen that minimizing the appearance term encourages the appearance coherence and disparity within and between the foreground and background regions, respectively. Note that, the minimum value of the appearance term is achieved when all the nodes are labeled as foreground or background, *i.e.,* $[x_A, x_B, x_c, x_D] = [0, 0, 0, 0]$ or $[1, 1, 1, 1]$. However, this rarely occurs in practice because this will cause a very high unary potential, while the overall objective is to minimize



$[x_A, x_B, x_C, x_D]$	$\Phi_a(\mathbf{s}, \mathbf{x}, \Theta)$
[0, 0, 0, 0]	0.00
[0, 0, 0, 1]	20.56
[0, 0, 1, 0]	20.56

$[x_A, x_B, x_C, x_D]$	$\Phi_a(\mathbf{s}, \mathbf{x}, \Theta)$
[0, 0, 1, 1]	79.31
[0, 1 0, 0]	20.56
[0, 1, 0, 1]	17.67

$[x_A, x_B, x_C, x_D]$	$\Phi_a(\mathbf{s}, \mathbf{x}, \Theta)$
[0, 1, 1, 0]	79.31
[0, 1 1, 1]	20.56

Fig. 3. An example illustrating how the potential term defined in Eq.(10) and Eq.(11) enforces the appearance constraints. The bar plots show the histogram features of four example superpixel nodes, and the tables show the cost incurred by labeling the four nodes differently.

the summation of all the three potential terms. The second best labeling in Fig. 3, *i.e.,* [0, 1, 0, 1] (or [1, 0, 1, 0]), implies that the first two bins mainly correspond to the background (or foreground), and the last two bins mainly correspond to the foreground (or background).

By substituting Eq.(11) into Eq.(10) we have

$$\begin{aligned}\Phi_a(\mathbf{s}, \mathbf{x}, \Theta) &= -\sum_{i}^{N}\sum_{j}^{M_i}\sum_{k=1}^{H}\delta(x_i^j, 1) \times b_i^{j,k} \times \ln p(F|k) \\ &\quad +\delta(x_i^j, 0) \times b_i^{j,k} \times \ln p(B|k) \\ &= -\sum_{k=1}^{H}\sum_{i}^{N}\sum_{j}^{M_i}\delta(x_i^j, 1) \times b_i^{j,k} \times \ln p(F|k) \\ &\quad +\delta(x_i^j, 0) \times b_i^{j,k} \times \ln p(B|k) \\ &= -\sum_{k=1}^{H}[\ln p(F|k) \times \sum_{i}^{N}\sum_{j}^{M_i}\delta(x_i^j, 1) \times b_i^{j,k} \\ &\quad + \ln p(B|k) \times \sum_{i}^{N}\sum_{j}^{M_i}\delta(x_i^j, 0) \times b_i^{j,k}] \\ &= -\sum_{k=1}^{H}\left(\Omega_F^k \times \ln\frac{\Omega_F^k}{\Omega_k} + \Omega_B^k \times \ln\frac{\Omega_B^k}{\Omega_k}\right). \quad (12)\end{aligned}$$

It can be seen that we arrive at similar conclusion as Eq.(8), and the new appearance term can also be equated by adding auxiliary nodes and edges to the original MRF structure. The difference is that we now add edges to connect every pair of superpixel and appearance auxiliary node, and the edge weight is set to the corresponding bin's vote. For example, the weight of the auxiliary edge connecting superpixel node s_i^j and the k^{th} auxiliary node is the number of feature points in s_i^j that are assigned to the k^{th} bin. This process is illustrated in Fig. 4. Compared with the original pixel wise approach, the proposed method is applicable to more complicated features besides color and can handle the cases where each node is described by a full histogram instead of a single bin.

A potential concern of the proposed framework is that the dimensionality of the histogram feature, *i.e.,* the number of auxiliary nodes need to be added, is extremely large due to the effect of Cartesian Product. For example, if we use 64 bins for each RGB channel, 100 words for both the

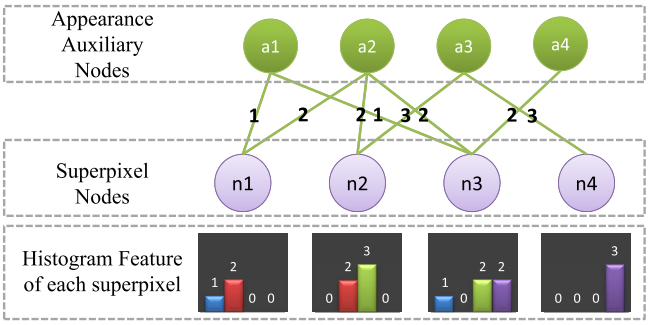


Fig. 4. A toy example illustrating how the appearance auxiliary nodes are connected to the superpixel nodes. In this example, there are only four superpixel nodes indicated by the purple discs. Each superpixel node is described by a 4-bin histogram which corresponds to the four green discs on the top. The numbers on the green edges indicate the weights of the auxiliary connections between the superpixel nodes and auxiliary nodes. Note that the edges between the superpixel nodes are omitted for simplicity.

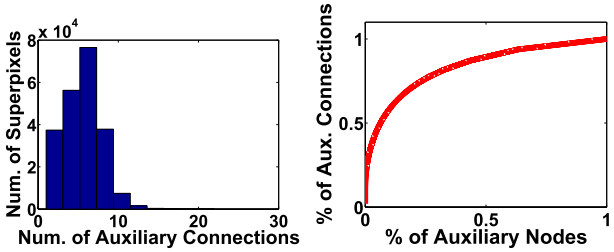


Fig. 5. The statistics on the amount of auxiliary connections linked to each superpixel node (left) and auxiliary node (right) on the bird_of_paradise sequence in SegTrack v2. The left plot shows the histogram on the number of auxiliary connections linked to each superpixel node. On the right plot, the horizontal axis is the percentage of auxiliary nodes and the vertical axis is the percentage of the total amount of auxiliary connections, *e.g.,* point (0.3, 0.8) means 30% of the auxiliary nodes with the highest connectivity contribute 80% of the auxiliary connections. Note that we choose not to simply plot the histogram on the number of auxiliary connections linked to each auxiliary node because the distribution is highly unbalanced.

dense SIFT and Texton bag of words features, there will be $64^3 \times 100 \times 100 \approx 2.6 \times 10^9$ bins in total. However, in practice, a superpixel node will be connected to an appearance auxiliary node only if the corresponding bin is not empty and an appearance auxiliary node will be added to the graph only when it is connected to at least two different superpixels. Hence, the actual number of auxiliary nodes and connections added to the graph is much smaller than the theoretical upper bound due to the sparsity of the histograms. For example, in a 98 frame video sequence, there are 221,559 superpixel nodes, 142,384 appearance auxiliary nodes and 1,105,807 connections between them. To show that the auxiliary connections are meaningfully distributed among the nodes, the statistics on the amount of auxiliary connections linked to each superpixel and auxiliary nodes are shown in Fig. 5 for the 98 frame video sequence. It can be seen that the auxiliary connections distribute stably among the superpixel nodes while highly unbalanced among the auxiliary nodes, *e.g.,* the mostly connected auxiliary node has around 21,570 connections while the least connected auxiliary node has only 2 connections. However, the mostly connected auxiliary node is far from dominating the auxiliary connections as it only contributes around 2% of the entire auxiliary connections.

TABLE I
COMPARISON RESULTS ON SEGTRACK V2 DATASET

video	video dimension	ours	ours w/o App.	[29]	[44]	[18]	[19]
bird_of_paradise	$640 \times 360 \times 98$	**94.49%**	76.12%	94.43%	-	92.20%	94.00%
birdfall2	$259 \times 327 \times 30$	66.23%	54.46%	57.78%	**71.00%**	49.00%	62.50%
frog	$480 \times 264 \times 279$	**80.68%**	47.16%	69.34%	74.00%	0.00%	65.80%
girl	$400 \times 320 \times 21$	81.63%	68.40%	74.94%	82.00%	87.70%	**89.20%**
monkey	$480 \times 270 \times 31$	68.51%	27.21%	64.02%	62.00%	79.00%	**84.80%**
monkeydog	$320 \times 240 \times 71$	**78.71%**	60.99%	78.19%	75.00%	-	58.80%
parachute	$414 \times 352 \times 51$	89.91%	60.36%	91.46%	94.00%	**96.30%**	93.40%
soldier	$528 \times 224 \times 32$	83.44%	64.78%	69.89%	60.00%	66.60%	**83.80%**
worm	$480 \times 364 \times 243$	81.57%	75.11%	74.19%	60.00%	**84.4%**	82.80%
average	-	**80.57%**	59.40%	74.92%	72.25%	69.40%	79.46%
runtime (seconds per frame)	-	6.84s	6.81s	14.90s	>82s	>82s	>82s

The video dimension is in the format of width×height×frame number.

D. Optimization

We use the max flow algorithm proposed in [7] to solve for the optimal labels. With the benefit of the proposed appearance modeling technique, the optimization is a single round process and it only takes seconds to optimize a video with hundreds of frames. As also shown in the experiment, the addition of the auxiliary nodes and edges only introduces negligible extra computation cost.

IV. EXPERIMENT

A. Dataset and Experimental Setup

In order to evaluate the effectiveness of the proposed appearance modelling technique, we run experiments on several benchmark datasets including the SegTrack v2[1] and 10-video-clip dataset[2] [13]. The videos in these two datasets are quite challenging. Many of the videos contain cluttered background and dynamic scenes due to camera motion or moving background objects. Some videos even contain fast motions such as the girl, monkey and monkeydog sequences in the SegTrack v2 dataset and the VWC102T, DO02_001 and DO01_055 sequences in ten video clip dataset. Some videos also contain cluttered background motions such as the swaying tree leaves and grass in the BR128T, BR130T and DO01_030 sequences in the ten video clip dataset. In some videos, the primary objects are visually very similar to the background, *i.e.,* low contrast along object boundaries, such as the birdfall, frog and worm sequences in the SegTrack v2 dataset. We evaluate the proposed approach against several state-of-the-art methods including both MRF based method [29] and non-MRF based methods [18], [19], [44]. We also compare with several baseline methods in order to separate the contributions of the different components. Pixel-wise Jaccard similarity coefficient, *i.e.,* intersection over union ratio, is used to evaluate the segmentation accuracy of each video.

The major parameters involved in the proposed method are the weights associated with each potential term in Eq.(2) and Eq.(6). In the experiment, we empirically set $\alpha_p\alpha_s = 240$, $\alpha_p\alpha_t = 160$, and $\alpha_a = 18$, and these parameters are kept fixed throughout all the experiments and videos unless otherwise specified. The β_s and β_t in Eq.(5) are set to the double average of the L2 feature distance between all the spatial and temporal pairs in a particular video, respectively, *i.e.,* $\beta_s = 2\langle\|\mathbf{F_i^j} - \mathbf{F_p^q}\|^2\rangle$ and $\beta_t = 2\langle\|\mathbf{H_i^j} - \mathbf{H_p^q}\|^2\rangle$ where $\langle.\rangle$ denotes averaging over all pairs. In the appearance modeling, we use 64 bins for each color channel and 100 words for both the dense SIFT and Texton histograms.

[1]http://www.cc.gatech.edu/~fli/SegTrack2/dataset.html
[2]http://www.brl.ntt.co.jp/people/akisato/saliency3.html

TABLE II
COMPARISON RESULTS ON TEN-VIDEO-CLIP DATASET

video	video dimension	ours	ours w/o App.	[29]
AN119T	$352 \times 288 \times 100$	**95.68%**	94.99%	94.50%
BR128T	$352 \times 288 \times 118$	**70.74%**	32.66%	34.82%
BR130T	$352 \times 288 \times 84$	**80.27%**	57.44%	29.17%
DO01_013	$352 \times 288 \times 89$	**93.84%**	79.74%	91.80%
DO01_014	$352 \times 288 \times 101$	93.62%	82.33%	**94.54%**
DO01_030	$352 \times 288 \times 101$	55.59%	18.24%	**77.91%**
DO01_055	$352 \times 288 \times 63$	52.53%	51.33%	**68.40%**
DO01_001	$352 \times 288 \times 83$	**93.22%**	39.15%	78.74%
M07058	$352 \times 288 \times 72$	81.16%	**82.95%**	77.71%
VWC102T	$352 \times 288 \times 107$	**83.72%**	78.09%	83.70%
average	-	**80.04%**	61.69%	73.13%

The video dimension is in the format of width×height×frame number.

B. Experimental Results

The comparison results with some state-of-the-art methods for both datasets are shown in Table I and II. Some qualitative comparisons are also shown in Fig. 6 and 7. From the numerical comparisons, it can be seen that the proposed method is not only faster but also more accurate than the existing state-of-the-art approaches for both datasets. The efficiency of the proposed method is because of its simplicity, *i.e.,* one graph cut on a sparsely connected graph in which the unary, pairwise and appearance potentials can be computed efficiently. The importance of appearance modeling is also revealed by comparing to our baseline approach without appearance constraint (the columns under "ours w/o App." in Table I and II). From the qualitative examples in Fig. 6, it can be seen that our initial saliency estimation is usually noisy and can only highlight the rough location of the primary object without detailed shape and boundary. As a consequence, our baseline approach without appearance constraint can only improve the segmentation performance by smoothing around the local edges. It is not able to correct those large regions corrupted by saliency. Moreover, the two examples shown in Fig. 7 imply that our method can handle the cases where the

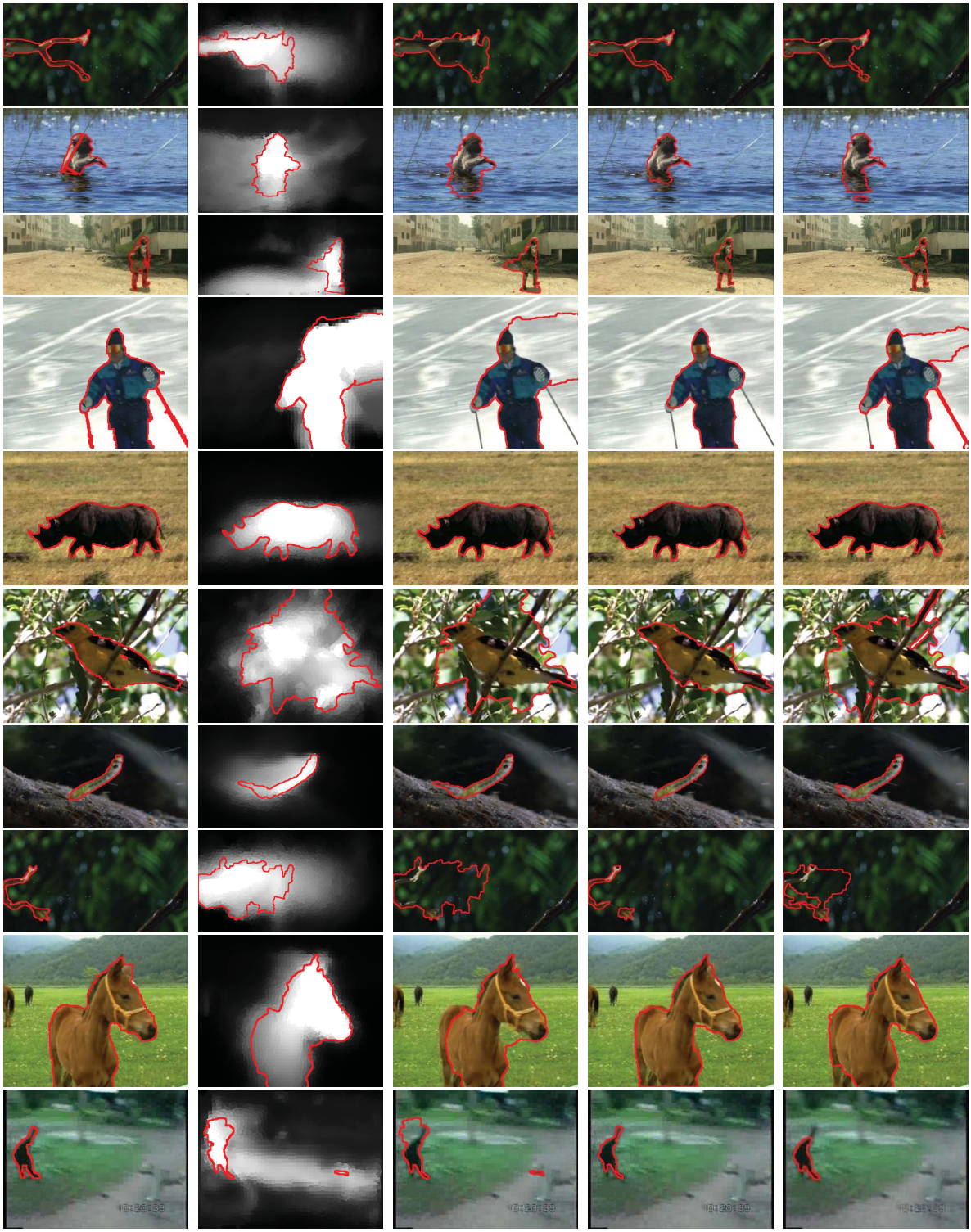


Fig. 6. Some qualitative results and comparisons.

primary object is absent in some frames. The method in [29] applies appearance constraint by training color GMMs in the local frames iteratively. It has shown better performance over our baseline approach but still fails when there is color overlap between foreground and background or the saliency estimation is consistently corrupted in a sequence of frames. Compared with [29], our appearance model is a global model across all the frames and employs more powerful features besides color. It consistently outperforms [29] in the shown examples. Furthermore, the addition of the appearance constraint only introduces negligible extra computation cost due to its efficiency.

Besides comparing with the state of the art, we also compare with several baseline methods in order to show the importance

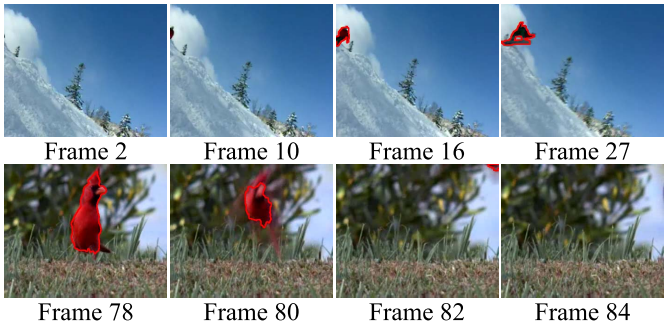


Fig. 7. Two examples in which the object is absent in the beginning (top example) or end (bottom example).

of the various components in the proposed method. The compared baseline methods are:

1) Segmentation by unary potential. In this appraoch, we exclude the pairwise and appearance terms. It directly measures the quality of the initial saliency estimation.
2) Naive extension of [35] (1). This approach applies the image based pixel wise segmentation method proposed in [35] to each individual frame. In this method, we compute the unary potential as in Section III-A, formulate the spatial pairwise potential based on the description in [32] and add the appearance constraint following [35]. The weights on the pairwise and appearance terms are set to 4 by grid search to accommodate the changes of the potential definitions.
3) Naive extension of [35] (2). In this approach, we use the average RGB value of each superpixel to describe each node. It directly applies the technique proposed in [35] since each node only corresponds to one bin in the color histogram space. The weight on the appearance term is reset to 7 by grid search to accommodate this change.
4) Our method without SIFT/Texton features, *i.e.,* ours with only color features. This baseline approach removes the dense SIFT and Texton features in the appearance modeling process. The difference to baseline (3) is that we still extract color features from sampled key points instead of computing the average. The weight on the appearance term are set to 2.7 by grid search to accommodate this change.

The comparison results in terms of the average Jaccard similarity coefficient for all the videos are shown in Fig. 8(a). Note that we have also used HSV color space in place of RGB in the baseline setting (3), (4) and the proposed full method, and the weights on the appearance terms are re-tuned for fair comparison. The paired t-tests with significance level of 0.05 have also been conducted to show the statistical meaningfulness of these comparisons. The p-values of these tests are shown in Fig. 8(b). From Fig. 8(b), it can be seen that the comparisons between baseline 1 and all the other methods and the comparisons between the proposed methods and all the baseline methods are statistically meaningful. The comparisons among all the rest baseline methods are not statistically meaningful. From the poor performance of baseline (1), it can be seen that the initial saliency estimation is far from a good segmentation. The comparison with baseline (4) shows the benefits of adding the dense SIFT and Texton feature by Cartesian Product. The comparisons with baseline (2) and (3) show that it is not trivial to extend the method proposed in [35] to videos and validate the necessity of our superpixel based approach with rich features.

C. Parameter Analysis

The major parameters involved in this framework are the three weights associated with the unary term, pairwise term and appearance term, respectively. Since the unary and pairwise terms have been explored in most of the MRF segmentation formulations, we evaluate the weight on the newly proposed appearance term in this section. In order to do this, we conduct experiment to compare the segmentation accuracy by varying this weight and the results for both datasets are shown in Fig. 9(a) and (b), respectively. It can be seen that, although each video sequence has its own preferred optimal weight, their trends are roughly consistent, *i.e.,* segmentation accuracy improves rapidly with increasing weights at the beginning, gradually saturates around 50 to 100 and some videos start to drop after 100. This implies that, within a wide range, the framework is not very sensitive to the weight on this newly proposed appearance term. We have also compared the segmentation accuracy between using an universal weight for all the videos as described in Section IV-A and individually selecting the best weight for each video. The result is shown in Fig. 9(c) and it can be seen that tunning the weight for each individual video can produce more accurate segmentations. However, the improvement is not very significant due to the stableness of the proposed technique on different videos, and an universal weight setting is generally more meaningful in practice.

D. Error Analysis

Despite the good performance of the proposed approach, segmentation errors are always inevitable and some typical examples are shown in Fig. 10. The most common error is the inclusion of background or exclusion of foreground along low contrast object boundaries, such as the left leg of the frog in the first column of Fig. 10, the right arm of the monkey in the second column of Fig. 10 and the reflection of the monkey on the water surface in the third column of Fig. 10. This is the built-in difficulty of primary object segmentation as we do not have prior knowledge of the object of interest and it is challenging to generate an accurate boundary in these low contrast regions. The second type of error is the inclusion of background regions in the gap between the object parts such as the grass between the two legs of the monkey in the second column of Fig. 10. These regions are labeled as foreground because they are blurred with high saliency value by the saliency warping/smoothing process along imperfect optical flows. The third type of error is the loss of thin structures attached to the main body of the object such as the legs of the bird in the last column of Fig. 10. These thin parts are either missed by the initial saliency estimation or smoothed away by the MRF smoothing. A common solution in the static image segmentation literature is to employ higher

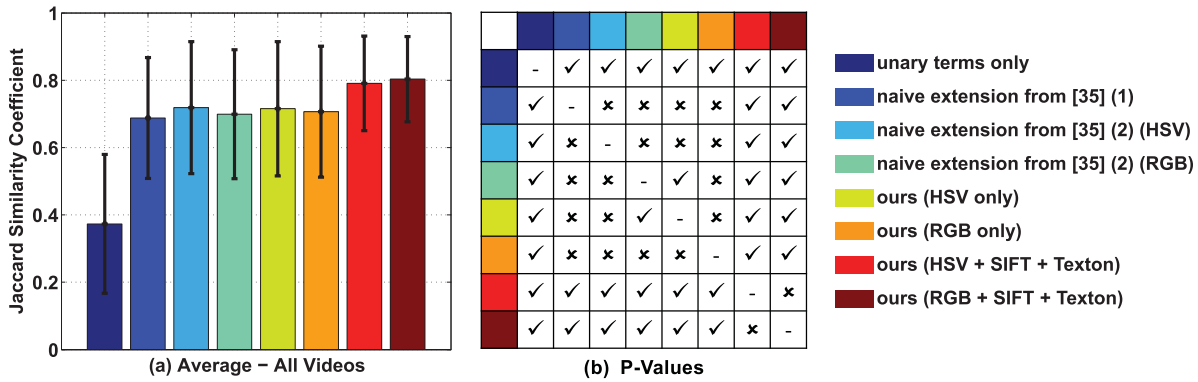


Fig. 8. Comparisons with several baseline methods. (a) shows the average segmentation accuracy of the proposed methods and several baseline settings. (b) shows the p-values of the paired t-tests conducted on the pairwise comparisons among the 8 methods listed in (a). A tick symbol means the p-value is smaller than 0.05 and a cross symbol means the p-value is greater than 0.05. It can be seen that, the comparisons between the proposed methods (last two bars in (a)) and the baseline methods (first 6 bars in (a)) are statistically meaningful.

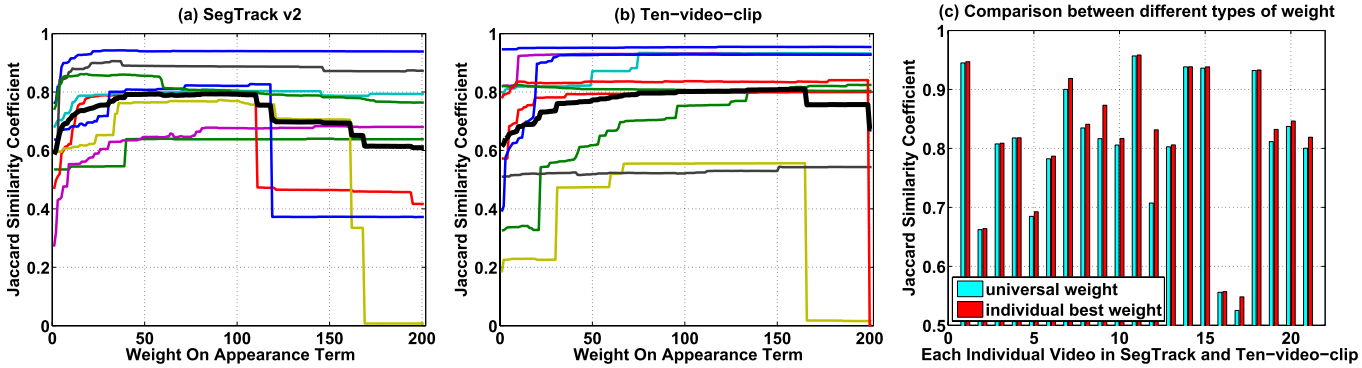


Fig. 9. Evaluation results regarding the weight on the appearance term. The first two curves show the segmentation accuracy of the SegTrack v2 and ten-video-clip dataset, respectively, by varying the weight from 0 to 200. The colorful thin lines indicate each individual video and the black thick lines indicate the average of each dataset. The right most bar plot shows the comparison of segmentation accuracy between using a universal weight for all the videos as described in Section IV-A and individually selecting the best weight for each video according to the first two curves. In the bar plot, horizontal label 1-9 indicate the 9 videos in the SegTrack v2 dataset, 10 indicates the average of SegTrack v2 dataset, 11-20 indicate the 10 videos in the ten-video-clip dataset and 21 indicates the average of the ten-video-clip dataset. Note that the vertical axis of the bar plot starts from 0.5 instead of 0.

Fig. 10. Some typical segmentation errors. The first row is the segmentation result, the second row is the ground truth segmentation and the last row is the segmentation errors.

order potentials, such as Robust P(n) [17], to enforce the high level structure of the object, and we leave this as our future work. In addition, the segmentation error in the fourth column of Fig. 10 is caused by severely corrupted saliency estimations. The saliency consistently fails to highlight the lower part of the flower due to the cluttered motion background, *e.g.*, both the

TABLE III
TIME USAGE OF THE VARIOUS COMPONENTS

Components	Runtime (seconds per frame)
amc saliency	0.22
gbmr saliency	1.25
optical flow	4.82
superpixel segmentation	0.15
SIFT feature extraction	0.80
Texton feature extraction	1.15
gc saliency	1.08
w saliency	0.47
saliency fusion	0.34
graph construction	0.58
MRF inference	0.01
total w/o parallelization	10.86
total with parallelization	6.84

flower and the leaves are swaying in the wind. Moreover, there happens to be a strong edge between the heart and the upper part of the flower and the MRF smoothing fails to prevent the separation.

E. Computation Speed

As shown in Table I, our method is efficient compared with the other approaches and the detailed time usage of the various components is shown in Table III. All the experiments are conducted on a Dual-Core i5 PC with 8GB of RAM, and the time statistics in Table III are based on the bird_of_paradise sequence in SegTrack v2 because it has the highest per-frame resolution. For the AMC [34][3] and GBMR [41][4] image saliency detection, SLIC [1][5] superpixel segmentation and structured forests edge detection [10],[6] we use the code provided by the authors. For optical flow computation, Texton feature extraction and MRF inference, we use the code provided with [7],[7] [22],[8] and [36],[9] respectively. For SIFT feature extraction, we use the VLFeat implementation.[10] All the other components are implemented by ourselves in Matlab. The detailed parameter settings of the various components can be found in Section IV-A. Due to the good architecture of our method, we are able to parallelize many of the components. For example, we could run the two image saliency detections, optical flow computation, SLIC superpixel segmentation, and SIFT/Texton feature extraction concurrently in multiple threads since they do not depend on each other. Similarly, we can also compute the two motion saliency maps concurrently in two threads after obtaining optical flows. In Table III, we highlight the components that can run concurrently using the same color. Overall, we can achieve 6.84 seconds[11] per frame with these two parallelization schemes.

[3]http://202.118.75.4/lu/Project/saliency_MC_iccv13/absorb_MC.html

[4]http://faculty.ucmerced.edu/mhyang/project/cvpr13_saliency/cvprsaliency.htm

[5]http://ivrl.epfl.ch/research/superpixels

[6]https://github.com/pdollar/edges

[7]http://pub.ist.ac.at/∼vnk/software.html

[8]https://people.csail.mit.edu/celiu/OpticalFlow/

[9]http://www.cs.unc.edu/∼jtighe/Papers/ECCV10/

[10]http://www.vlfeat.org/overview/dsift.html

[11]$6.84 = \max\{0.22, 1.25, 4.82, 0.80, 1.15, 0.15\} + \max\{1.08, 0.47\} + 0.34 + 0.58 + 0.01 = 4.82 + 1.08 + 0.34 + 0.58 + 0.01$. These numbers correspond to the entries in Table III.

From Table III it can be seen that the efficiency bottleneck of our method is the optical flow computation, saliency estimation and feature extraction. The graph construction and inference only contribute 5% of the total computational time. Hence, the efficiency of our method can be further improved with the recent advancement in GPU accelerated optical flow, *e.g.,* 0.2 second per frame in [6]. In the compared methods, [18], [19], [44] are significantly slower because they employ the more advanced but time consuming region proposals [8], [11] as the primitive input.

V. CONCLUSION

In this paper, we propose an efficient and effective appearance modeling technique in the MRF framework for automatic primary video object segmentation. The proposed method uses histogram features to characterize the local regions and embed the global appearance constraint into the graph by auxiliary nodes and connections. Compared with many existing appearance models, the optimization process of our method is non-iterative. Experimental evaluations show that our method is faster than many of the alternatives and the segmentation accuracy is also better than or comparable with the state-of-the-art methods.

REFERENCES

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

[2] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2189–2202, Nov. 2012.

[3] V. Badrinarayanan, F. Galasso, and R. Cipolla, "Label propagation in video sequences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3265–3272.

[4] 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, p. 70, 2009.

[5] D. Banica, A. Agape, A. Ion, and C. Sminchisescu, "Video object segmentation by salient segment chain composition," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Dec. 2013, pp. 283–290.

[6] L. Bao, Q. Yang, and H. Jin, "Fast edge-preserving patchmatch for large displacement optical flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3534–3541.

[7] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1124–1137, Sep. 2004.

[8] J. Carreira and C. Sminchisescu, "CPMC: Automatic object segmentation using constrained parametric min-cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1312–1328, Jul. 2012.

[9] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 569–582, Mar. 2015.

[10] P. Dollár and C. L. Zitnick, "Fast edge detection using structured forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 8, pp. 1558–1570, Aug. 2015.

[11] I. Endres and D. Hoiem, "Category independent object proposals," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 575–588.

[12] H. Fu, D. Xu, B. Zhang, and S. Lin, "Object-based multiple foreground video co-segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3166–3173.

[13] K. Fukuchi, K. Miyazato, A. Kimura, S. Takagi, and J. Yamato, "Saliency-based video segmentation with graph cuts and sequentially updated priors," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jun./Jul. 2009, pp. 638–641.

[14] M. Grundmann, V. Kwatra, M. Han, and I. Essa, "Efficient hierarchical graph-based video segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2141–2148.

[15] S. D. Jain and K. Grauman, "Supervoxel-consistent foreground propagation in video," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 656–671.

[16] C. Jung and C. Kim, "A unified spectral-domain approach for saliency detection and its application to automatic object segmentation," *IEEE Trans. Image Process.*, vol. 21, no. 3, pp. 1272–1283, Mar. 2012.

[17] P. Kohli, L. Ladický, and P. H. S. Torr, "Robust higher order potentials for enforcing label consistency," *Int. J. Comput. Vis.*, vol. 82, no. 3, pp. 302–324, 2009.

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

[19] 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. Vis.*, Dec. 2013, pp. 2192–2199.

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

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

[22] C. Liu, "Beyond pixels: Exploring new representations and applications for motion analysis," M.S. thesis, Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, 2009.

[23] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[24] Y. Luo, J. Yuan, P. Xue, and Q. Tian, "Saliency density maximization for efficient visual objects discovery," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 12, pp. 1822–1834, Dec. 2011.

[25] Y. Luo, G. Zhao, and J. Yuan, "Thematic saliency detection using spatial-temporal context," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Dec. 2013, pp. 347–353.

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

[27] F. Manerba, J. Benois-Pineau, R. Leonardi, and B. Mansencal, "Multiple moving object detection for fast video content description in compressed domain," *EURASIP J. Adv. Signal Process.*, vol. 2008, p. 5, Jan. 2008.

[28] C. Marc, P. Stéphane, and N. Henri, "Segmentation of non-rigid video objects using long term temporal consistency," in *Proc. Int. Conf. Image Process.*, vol. 2. 2002, pp. II-93–II-96.

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

[30] B. L. Price, B. S. Morse, and S. Cohen, "LIVEcut: Learning-based interactive video segmentation by evaluation of multiple propagated cues," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 779–786.

[31] P. Rantalankila, J. Kannala, and E. Rahtu, "Generating object segmentation proposals using global and local search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2417–2424.

[32] C. Rother, V. Kolmogorov, and A. Blake, "'GrabCut': Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, 2004.

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

[34] J. Sun, H. Lu, and X. Liu, "Saliency region detection based on Markov absorption probabilities," *IEEE Trans. Image Process.*, vol. 24, no. 5, pp. 1639–1649, May 2015.

[35] M. Tang, L. Gorelick, O. Veksler, and Y. Boykov, "GrabCut in one cut," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1769–1776.

[36] J. Tighe and S. Lazebnik, "SuperParsing: Scalable nonparametric image parsing with superpixels," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 638–641.

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

[38] M. Van den Bergh, X. Boix, G. Roig, B. de Capitani, and L. Van Gool, "Seeds: Superpixels extracted via energy-driven sampling," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 13–26.

[39] L. Wang, G. Hua, R. Sukthankar, J. Xue, and N. Zheng, "Video object discovery and co-segmentation with extremely weak supervision," in *Proc. 13th Eur. Conf. Comput. Vis.*, 2014, pp. 640–655.

[40] C. Xu and J. J. Corso, "Evaluation of super-voxel methods for early video processing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1202–1209.

[41] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3166–3173.

[42] J. Yang *et al.*, "Discovering primary objects in videos by saliency fusion and iterative appearance estimation," *IEEE Trans. Circuits Syst. Video Technol.*, to be published.

[43] J. Yuan, G. Zhao, Y. Fu, Z. Li, A. K. Katsaggelos, and Y. Wu, "Discovering thematic objects in image collections and videos," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2207–2219, Apr. 2012.

[44] 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. Vis. Pattern Recognit.*, Jun. 2013, pp. 628–635.

[45] D. Zhang, O. Javed, and M. Shah, "Video object co-segmentation by regulated maximum weight cliques," in *Proc. 13th Eur. Conf. Comput. Vis.*, 2014, pp. 551–566.

[46] G. Zhao, J. Yuan, and G. Hua, "Topical video object discovery from key frames by modeling word co-occurrence prior," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1602–1609.

[47] G. Zhao, J. Yuan, G. Hua, and J. Yang, "Topical video object discovery from key frames by modeling word co-occurrence prior," *IEEE Trans. Image Process.*, to be published.

[48] L. Zheng, S. Wang, Z. Liu, and Q. Tian, "Packing and padding: Coupled multi-index for accurate image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1947–1954.

Jiong Yang received the B.Eng. (Hons.) degree from the School of Electrical Electronic Engineering, Nanyang Technological University, Singapore, in 2013, where he is currently pursuing the Ph.D. degree with the Rapid Rich Object Search Laboratory, College of Engineering. His research interests include computer vision and machine learning.

Brian Price received the Ph.D. degree in computer science from Brigham Young University, Provo, UT, USA, under the advisement of Dr. B. Morse.

He has contributed new features to many Adobe products, such as Photoshop, Photoshop Elements, and After Effects, mostly involving interactive image segmentation and matting, as a Researcher with Adobe Research, San Jose, CA, USA. He is currently a Senior Research Scientist with Adobe Research specializing in computer vision. His research interests include semantic segmentation, interactive object selection and matting in images and videos, stereo and RGBD, image processing, and computer vision and its intersections with machine learning and computer graphics.

Xiaohui Shen received the B.S. and M.S. degrees from the Department of Automation, Tsinghua University, Beijing, China, in 2005 and 2008, respectively, and the Ph.D. degree from the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL, USA, in 2013.

He is currently a Research Scientist with Adobe Research, San Jose, CA, USA. His research interests include image/video processing and computer vision.

Zhe Lin received the B.Eng. degree in automatic control from the University of Science and Technology of China, Hefei, China, in 2002, the M.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology, Daejeon, Korea, in 2004, and the Ph.D. degree in electrical and computer engineering from the University of Maryland, College Park, MD, USA, in 2009.

He has been a Research Intern with Microsoft Corporation, Bellevue, WA, USA. He is currently a Senior Research Scientist with Adobe Research, San Jose, CA, USA. His research interests include deep learning, object detection and recognition, image classification, content-based image and video retrieval, human motion tracking, and activity analysis.

Junsong Yuan received the B.Eng. degree from the Special Class for the Gifted Young, Huazhong University of Science and Technology, the M.Eng. degree from the National University of Singapore, and the Ph.D. degree from Northwestern University. He is currently an Associate Professor and the Program Director of video analytics with the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore.

His research interests include computer vision, video analytics, gesture and action analysis, and large-scale visual search and mining. He serves as the Program Co-Chair of the IEEE Conference on Visual Communications and Image Processing (2015), the Organizing Co-Chair of the Asian Conference on Computer Vision (ACCV'14), and the Area Chair of the IEEE Winter Conference on Computer Vision (2014), the IEEE Conference on Multimedia Expo (2014 and 2015), and ACCV'14. He co-chairs six workshops at the IEEE Conference on Computer Vision (2013), SIGGRAPH Asia'14, and the IEEE Conference on Computer Vision and Pattern Recognition (2012, 2013, and 2015). He serves as a Guest Editor of the *International Journal of Computer Vision* and an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGIES and *The Visual Computer journal*.

He received Nanyang Assistant Professorship and Tan Chin Tuan Exchange Fellowship from Nanyang Technological University, Outstanding EECS Ph.D. Thesis award from Northwestern University, Doctoral Spotlight Award from CVPR'09, and National Outstanding Student from Ministry of Education, P.R.China. He is a senior member of IEEE.