Fast Appearance Modeling for Automatic Primary Video Object Segmentation

Jiong Yang, Brian Price, Xiaohui Shen, Zhe Lin and Junsong Yuan

***Abstract*—Automatic segmentation of the primary object in a video clip is a challenging problem as there is no prior knowledge of the foreground 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 initiation and can be easily trapped at local optimal. Also, 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. Extensive experimental evaluations validate the superiority of the proposed method over the state of the arts, in both efficiency and effectiveness.**

***Index Terms*—automatic, primary, video, object, segmentation, graph cut**

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. 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, *i.e.,* interactive segmentation [15], [4] and fully automatic segmentation [18], [41], [27], [20]. Our method belongs to the latter and does not assume the object is present in all the frames. Note that, throughout the paper we will use the terms “primary object”, “foreground object” or simply “foreground” interchangeably.

Following the outstanding performance of Markov Random Field (MRF) based method in image object segmentation [30], [33], [9], many of the existing video object segmentation approaches also build spatio-temporal MRF graph and show promising results [27], [15], [41]. These approaches build spatio-temporal graph by connecting spatially or temporally connected regions, *e.g.,* pixels [33] or superpixels [27], and cast the segmentation problem into a node labeling problem in a Markov Random Field. This process is illustrated graphically in Fig. 2. These techniques usually have three major components in the context of automatic primary video object segmentation: 1) Initial visual or motion saliency estimation; 2) Spatio-temporal graph connecting neighborhood regions; 3) Foreground and 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. Moreover, with the presence of appearance constraint, there are two groups of parameters in the optimization process, *i.e.,* segmentation labels $\mathbf{x}$ and appearance model $\mathbf{\Theta}$. It is intractable to solve both parameters simultaneously for many commonly used appearance models such as Gaussian Mixture Models (GMM) [27] or Multiple Instance Learning [37]. Hence, many existing methods adapt an iterative approach. They use the segmentation result of the k^{th} iteration to train foreground and background appearance models which are then used to refine the segmentation in the $(k+1)^{th}$ iteration. However, these methods can be easily trapped at local optimal and are time consuming especially for video data.

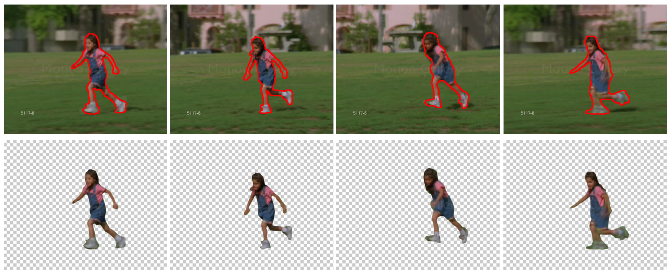

Fig. 1. Illustration of primary object segmentation in videos. The top row is the original video frames with the expected segmentation results rendered as contours. The bottom row is the same segmentation result after removing background.

Recently, [33] 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 some auxiliary nodes and edges into the original MRF structure. However, due to the fundamental difference between image data and video data, the original approach in [33] is not practically applicable to videos because it requires each node to be described by a single bin in the histogram space. For video object segmentation, superpixels are usually preferred due to large data volume and more robust features like SIFT [23] or Textons are generally used in addition to the raw color to better capture the viewpoint and lighting variations between different frames. As a result, each pixel will have multiple features and each node will correspond to multiple pixels. Hence, in this paper, we extend the efficient appearance modeling technique in [33] to primary video object segmentation by addressing these challenges. The proposed appearance modeling technique is more general than [33] and can handle all the mentioned difficulties. The resultant auxiliary connections are also different, *i.e.,* in [33] 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 the original approach.

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 can be solved efficiently by one graph cut. Although inspired by the idea of [33], we have made non-trivial extension from static images to videos, and 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 to the recent state of 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], [36] and supervoxel segmentation [38], [14]. Superpixel segmentation 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 the object level segmentation instead of those unsupervised low level pixel grouping methods. Instead, the superpixels and supervoxels are usually used as the primitive elements in place of pixels in the context of video object segmentation for efficiency [27], [37], [15]. Another type of low level segmentation is object proposal segmentation [11], [8], [29], which becomes popular in recent years. 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 to the superpixel or supervoxel methods. Many high level video object segmentation methods use the proposals as the primitive input [18], [19], [26], [41], [12], [42].

B. Object Level Video Segmentation

The existing works related to video object segmentation 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 Introduction, 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 [15], [3], [28], [35] 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 [27], [18], [26], [5], [19], [41]. 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 [18], [19], [26], [41], [5] use object proposals [11] as the primitive input which contributes a significant portion to the inefficiency of these methods. In [18], it first uses spectral clustering to group proposals with coherent appearance and then train foreground/background color GMMs and foreground object location priors. Pixel-wise graph cut is used to produce the final segmentation mask for each individual frame. In [26], it adapts a similar pipeline of [18] but uses constrained maximum weighted clique to group proposals. In [41], it builds spatial-temporal graph by connecting proposals and uses dynamic programming to find the most confident trajectory. They then use pixel-wise graph cut to refine the final segmentation mask for each individual frame based on the initial proposal trajectory. In [5], it produces multiple proposal chains by linking local segments using long-range temporal constraints. It also learns the location prior and GMM based foreground prior from the coarse chains and refine the final segmentation result by pixel-wise per-frame MRF smoothing. In [19], it tracks the proposals temporally using incremental regression and refine the final segmentations by composite statistic inferences. In [25], it first segments the selected key frames into an over complete set of segments using image segmentation algorithms like [31] and then employs the cohesive sub-graph mining technique to find the salient segments with similar appearance and strong mutual affinity. [43] adapts a similar pipeline but uses topic model to discover the coherent segments. Both of them 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], [37], [42]. Both [12] and [37] formulate the segmentation as node selection or labeling in spatio-temporal graph while [42] finds the maximum weighted

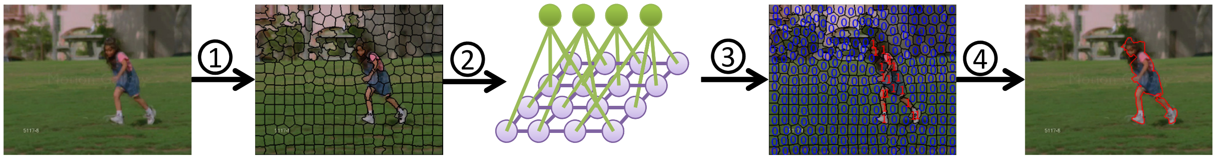

Fig. 2. The overall work flow of the proposed segmentation framework. 1. Superpixel segmentation; 2. Graph construction: the purple nodes and edges represent the superpixels and the spatio-temporal neighborhood connections between them. They are used to encourage the spatio-temporal smoothness of the 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.

clique in a completely connected graph. In [12], it does not have an explicit global appearance model and [37] 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 [27], [30], [15], [26], [18], [41], [5]. Multiple instance learning on context features is also used in [37] 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, [33] proposed to use histogram features to model the appearance in a non-parametric manner for static image segmentation. Both the appearance model and segmentation labels can be globally 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. 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

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

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

The vector $\mathbf{x}$ and $\boldsymbol{\Theta}$ denote the $\{0,1\}$ labels 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.

A. Unary Potentials

Since saliency has been proven to be effective in highlighting the foreground object in a completely automatic setting [2], [16], [24], [27], 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 [32], GBMR image saliency, [39], GC motion saliency [40] and W motion saliency [40]. The four types of saliency maps are fused by the adaptive SVM-Fusion technique [40] to produce a single saliency map for each frame. We also warp the saliency estimations along the optical flow direction to encourage temporal smoothness. If we use $A(s_i^j)$ to denote the average saliency value of superpixel s_i^j, its unary potential is given by:

$$\phi_u(s_i^j) = \begin{cases} -\log(A(s_i^j)) & \text{if } x_i^j = 1 \\ -\log(1 - A(s_i^j)) & \text{if } x_i^j = 0 \end{cases}. \tag{2}$$

Then the total unary term in Eq.(1) can be computed as

$$\Phi_u(\mathbf{s},\mathbf{x}) = \sum_i^N \sum_j^{M_i} \phi_u(s_i^j). \tag{3}$$

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 neighborhood and temporal neighborhood. 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 neighborhood superpixels will have nonzero edge and the edge weight represents the cost induced by assigning different labels to the 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 result on both the appearance and motion domain to determine the edge weight. More specifically, we use color and optical flow orientation histogram to compare the local similarity and the structural forest edge detector [10] to compute the spatio-temporal edge strengths. Note that, to detect motion edge for each frame, we first convert the XY dense flow vector of each pixel to RGB colors using the method in [22] and then apply the edge detector in the RGB domain. These two edge maps, *i.e.,* appearance and motion, are then combined by maximum operation. Overall, the spatial and temporal pairwise potentials between connected superpixels are computed as

$$\begin{aligned}\phi_s(s_i^j, s_p^q) &= (1 - e(s_i^j, s_p^q))\delta[x_i^j, x_p^q]\exp(-\beta_s^{-1}\|\mathbf{F_i^j} - \mathbf{F_p^q}\|^2)\\ \phi_t(s_i^j, s_p^q) &= c(s_i^j, s_p^q)\delta[x_i^j, x_p^q]\exp(-\beta_t^{-1}\|\mathbf{I_i^j} - \mathbf{I_p^q}\|^2). \qquad (4)\end{aligned}$$

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 $\delta[.]$ is a discrete indicator function where $\delta[u, v] = 0$ if $u = v$ and $\delta[u, v] = 1$ if $u \neq v$. $\mathbf{F_i^j}$ is the concatenation of color and optical flow orientation histogram and $\mathbf{I_i^j}$ is the color histogram. The motion feature is only included in the spatial pairwise potentials because temporal edges connect superpixels in different frames. The overall pairwise potential is then computed as the weighted summation of all the spatial and temporal pairwise terms:

$$\Phi_p(\mathbf{s}, \mathbf{x}) = \alpha_s \sum_{\{s_i^j, s_p^q\} \in \mathcal{N}_s} \phi_s(s_i^j, s_p^q) + \alpha_t \sum_{\{s_i^j, s_p^q\} \in \mathcal{N}_t} \phi_t(s_i^j, s_p^q) \qquad (5)$$

where $\mathcal{N}_s$ and $\mathcal{N}_t$ denote the collections of all the spatial and temporal neighborhood pairs, respectively.

C. Appearance Auxiliary Potential

Since the proposed appearance modeling technique is inspired by [33], we first review their method on static image segmentation and then discuss the challenges in adapting the idea to videos and how we overcome these challenges.

In [33], they model each pixel as a node and represent each node as a single bin in the RGB histogram space for appearance modeling. Let us use p_i and x_i to denote the i^{th} pixel and its label, respectively, b_i to denote the assigned bin of pixel p_i, H to denote the dimensionality of the histogram space and P to denote the total number of pixels. Furthermore we use Ω_F^k and Ω_B^k to denote the number of pixels in the foreground and background regions, respectively, that are assigned to the k^{th} bin, *i.e.,* $\Omega_F^k = |\{p_i | x_i = 1\}|$ and $\Omega_B^k = |\{p_i | x_i = 0\}|$ where $|.|$ denotes the Cardinality of a set, Ω^k to denote the number of pixels in the entire image that are assigned to the k^{th} bin, *i.e.,* $\Omega^k = \Omega_F^k + \Omega_B^k$. Then the foreground and background probability of the k^{th} histogram bin is given by $p(F|k) = \frac{\Omega_F^k}{\Omega^k}$ and $p(B|k) = \frac{\Omega_B^k}{\Omega^k}$, respectively. Finally, the appearance constraint potential of each pixel p_i can be computed as

$$\phi_a(p_i) = \begin{cases} -\ln p(F|b_i) \text{if } x_i = 1 \\ -\ln p(B|b_i) \text{if } x_i = 0 \end{cases} . \qquad (6)$$

Then the total appearance constraint potential of all the pixels, *i.e.,* the equivalence of the last term in Eq.(1), can be computed as

$$\begin{aligned}\Phi_a &= \sum_{i=1}^{P} \phi_a(p_i)\\ &= \sum_{i=1}^{P} -\delta[x_i, 0]\ln p(F|b_i) - \delta[x_i, 1]]\ln p(B|b_i)\\ &= -\sum_{i=1}^{P}(\delta[x_i, 0]\ln\frac{\Omega_F^{b_i}}{\Omega^{b_i}} + \delta[x_i, 1]\ln\frac{\Omega_B^{b_i}}{\Omega^{b_i}})\\ &= -(\sum_{k=1}^{H}\Omega_F^k\ln\frac{\Omega_F^{b_i}}{\Omega^{b_i}} + \sum_{k=1}^{H}\Omega_B^k\ln\frac{\Omega_B^{b_i}}{\Omega^{b_i}})\\ &= -\sum_{k=1}^{H}(\Omega_F^k\ln\frac{\Omega_F^{b_i}}{\Omega^{b_i}} + \Omega_B^k\ln\frac{\Omega_B^{b_i}}{\Omega^{b_i}}). \qquad (7)\end{aligned}$$

The inner part of the summation in Eq.(7) can be approximated by $|\Omega_F^k - \Omega_B^k|$ since $\Omega_F^k + \Omega_B^k = \Omega^k$. Hence, $\Phi_a(\mathbf{x}, \mathbf{\Theta}) \approx -\sum_{k=1}^{H}|\Omega_F^k - \Omega_B^k| = \sum_{k=1}^{H} 2\min(\Omega_F^k, \Omega_F^k) - \Omega^k$. As we are only interested in minimizing $\Phi_a(\mathbf{x}, \mathbf{\Theta})$ instead of its absolute value, we can drop the constant term Ω^k and the multiplier 2. Eventually, the appearance model is reduced to

$$\Phi_a(\mathbf{x}, \mathbf{\Theta}) = \sum_{k=1}^{H}\min(\Omega_F^k, \Omega_F^k) \qquad (8)$$

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.

An naive extension of [33] 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 well 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 [34], [40], [44], [25]. However both the original appearance modeling approach and the naive extension are not readily applicable to these multi-feature situations because both SIFT and Texton are key point based features and are not confined to any arbitrarily shaped superpixels. 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 take their Cartesian Product to assign each feature point to a single bin. However, unlike the case of single pixels, each node in our superpixel based approach may contain more than one feature point and the original approach in [33] is not directly applicable here. Hence, we propose a variation of the original technique that can 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. As proven below, we achieve similar theoretical conclusion but are applicable to more general cases.

For consistency, we first redefine some of the terms used in the description of the pixel wise approach in [33]. 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 assigned to the k^{th} bin in superpixel s_i^j, 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, *i.e.,* $\Omega_F^k = \sum_i^N \sum_{j,x_i^j=1}^{M_i} b_i^{j,k}$, $\Omega_B^k = \sum_i^N \sum_{j,x_i^j=0}^{M_i} b_i^{j,k}$, and Ω^k denote the total number of votes in all the superpixels' k^{th} bin, *i.e.,* $\Omega^k = \Omega_F^k + \Omega_B^k$. 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.(1) is defined as $\Phi_a(\mathbf{x}, \mathbf{\Theta}) = \sum_i^N \sum_j^{M_i} \phi_a(s_i^j)$ where

$$\begin{aligned}\phi_a(s_i^j) &= \begin{cases} -\ln p(F|s_i^j) & \text{if } x_i^j = 1 \\ -\ln p(B|s_i^j) & \text{if } x_i^j = 0 \end{cases} \\ &= \begin{cases} -\sum_{k=1}^H b_i^{j,k} \ln p(F|k) & \text{if } x_i^j = 1 \\ -\sum_{k=1}^H b_i^{j,k} \ln p(B|k) & \text{if } x_i^j = 0 \end{cases}\end{aligned} \tag{9}$$

and $\Phi_a(\mathbf{x}, \mathbf{\Theta})$

$$\begin{aligned}&= -\sum_i^N \sum_j^{M_i} \sum_{k=1}^H \delta[x_i^j, 0] b_i^{j,k} \ln p(F|k) + \delta[x_i^j, 1] b_i^{j,k} \ln p(B|k) \\ &= -\sum_{k=1}^H \sum_i^N \sum_j^{M_i} \delta[x_i^j, 0] b_i^{j,k} \ln p(F|k) + \delta[x_i^j, 1] b_i^{j,k} \ln p(B|k) \\ &= -\sum_{k=1}^H [\ln p(F|k) \sum_i^N \sum_j^{M_i} \delta[x_i^j, 0] b_i^{j,k} + \\ &\quad \ln p(B|k) \sum_i^N \sum_j^{M_i} \delta[x_i^j, 1] b_i^{j,k}] \\ &= -\sum_{k=1}^H \left(\Omega_F^k \ln \frac{\Omega_F^k}{\Omega_k} + \Omega_B^k \ln \frac{\Omega_B^k}{\Omega_k} \right).\end{aligned} \tag{10}$$

It can be seen that we arrive at similar conclusion as Eq.(7). This is also equivalent to adding some auxiliary nodes and edges to the original MRF structure. The difference is that, we now add edges to connect each pair of the superpixel and appearance auxiliary node and the edge weight is set to the corresponding bin's vote, *i.e.,* the weights of the auxiliary edge connecting superpixel node s_i^j and the k^{th} auxiliary node is the number of feature points assigned to the k^{th} bin in s_i^j. This process is illustrated in Fig. 3. Compared to the original pixel wise approach, the proposed variation is applicable to more complicated features besides raw color and can handle the cases where each node is described by a full histogram instead of a single bin.

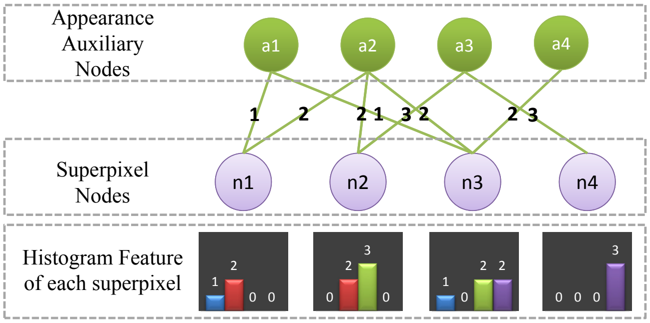
Appearance Auxiliary Nodes
a1
a2
a3
a4
1
2
2
1
3
2
2
3
Superpixel Nodes
n1
n2
n3
n4
Histogram Feature of each superpixel
1
2
0
0
0
2
3
0
1
0
2
2
0
0
0
3

Fig. 3. 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 correspond 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.

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 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 non empty and an appearance auxiliary node will be added to the graph only if it is connected to at least two different superpixels. Hence, the actual number of auxiliary nodes and connections added to the graph is much less than the theoretical upper bound due to the sparsity of the histograms. For example, in a 98 frame video sequence, there are 221559 superpixel nodes, 142384 appearance auxiliary nodes and 1105807 connections between them. To show that the auxiliary connections are properly distributed among the nodes, the statistics on the amount of auxiliary connections linked to each superpixel and auxiliary nodes are shown in Fig. 4 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 most connected auxiliary node has around 21570 connections while the least connected auxiliary node has only 2 connections. However, the most connected auxiliary node is far from dominating the auxiliary connections as it only contributes around 2% of the entire auxiliary connections.

D. Optimization

We use the max flow algorithm proposed in [7] to solve for the optimal labels. With the benefit of the proposed appearance

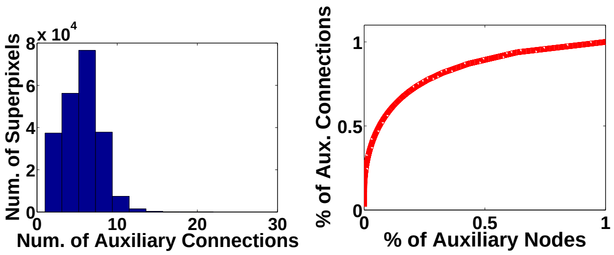

Fig. 4. The statistics on the amount of auxiliary connections linked to each superpixel node (left) and auxiliary node (right) on the bird_of_paradise sequence in SegTrack v2. The left plot shows the histogram on the number of auxiliary connections linked to each superpixel node. On the right plot, the 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 to not simply plot the histogram on the number of auxiliary connections linked to each auxiliary node because the distribution is highly unbalanced.

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 part, the addition of the auxiliary nodes and edges only introduces negligible extra computation cost.

IV. Experiment

A. Dataset Experimental Setup

In order to evaluate the effectiveness of the proposed appearance modeling technique, we run experiments on several benchmark datasets including the SegTrack v2 and 10-video-clip dataset [13]. We evaluate the proposed approach against several state-of-the-art methods including both MRF based method [27] and non-MRF based methods [41], [19], [18]. We also compare with several baseline methods in order to show the importance of the various components. Pixel-wise overlap 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.(1) and Eq.(5). In the experiment, we empirically set $\alpha_p \alpha_s = 240$, $\alpha_p \alpha_t = 160$, and $\alpha_a = 16$. These parameter are kept fixed throughout all the experiments and videos unless otherwise specified. The β_s and β_t in Eq.(4) 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 $\beta_t = 2\langle\|\mathbf{I_i^j} - \mathbf{I_p^q}\|^2\rangle$ where $\langle.\rangle$ denotes averaging over all pairs. In the appearance modeling, we use 64 bins for each RGB channel and 100 words for both the dense SIFT and Texton histograms.

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. 5. 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 both the unary and pairwise potentials can be computed efficiently. The importance of appearance modeling is also revealed by comparing to our baseline approach without appearance constraint. From the qualitative examples in Fig. 5, 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 appearance without appearance constraint can only improve the segmentation performance by smoothing around the local edges. It will not be able to correct those large regions corrupted by saliency. The method in [27] 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 to [27], our appearance model is a global model across all the frames and employs more powerful features besides color. It consistently outperforms [27] in the shown examples. Furthermore, the addition of the appearance constraint only introduces negligible extra computation cost due to its efficiency.

TABLE II
Comparison results on ten-video-clip dataset

video	video dimension	ours	ours w/o App.	[27]
AN119T	352 × 288 × 100	95.68%	94.99%	95.54%
BR128T	352 × 288 × 118	70.74%	32.66%	21.78%
BR130T	352 × 288 × 84	80.27%	57.44%	24.73%
DO01_013	352 × 288 × 89	93.84%	79.74%	81.17%
DO01_014	352 × 288 × 101	93.62%	82.33%	92.83%
DO01_030	352 × 288 × 101	55.59%	18.24%	72.95%
DO01_055	352 × 288 × 63	52.53%	51.33%	76.30%
DO01_001	352 × 288 × 83	93.22%	39.15%	62.31%
M07058	352 × 288 × 72	81.16%	82.95%	79.60%
VWC102T	352 × 288 × 107	83.72%	78.09%	38.58%
average	-	80.04%	61.69%	64.58%

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

Besides comparing with the state of the art, we also compare with several baseline methods in order to show the importance 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) Direct extension of [33] (1). This approach applies the image based pixel wise segmentation method proposed in [33] 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 [30] and add the appearance constraint following [33]. The weights on the pairwise and appearance terms are set to 4, respectively, by grid search to accommodate the changes of the potential definitions.
3) Direct extension of [33] (2). In this approach, we use the average RGB value of each superpixel to describe each node. It directly applies the technique proposed in [33] since each node only corresponds to one bin in the histogram space. The weight on the appearance term is

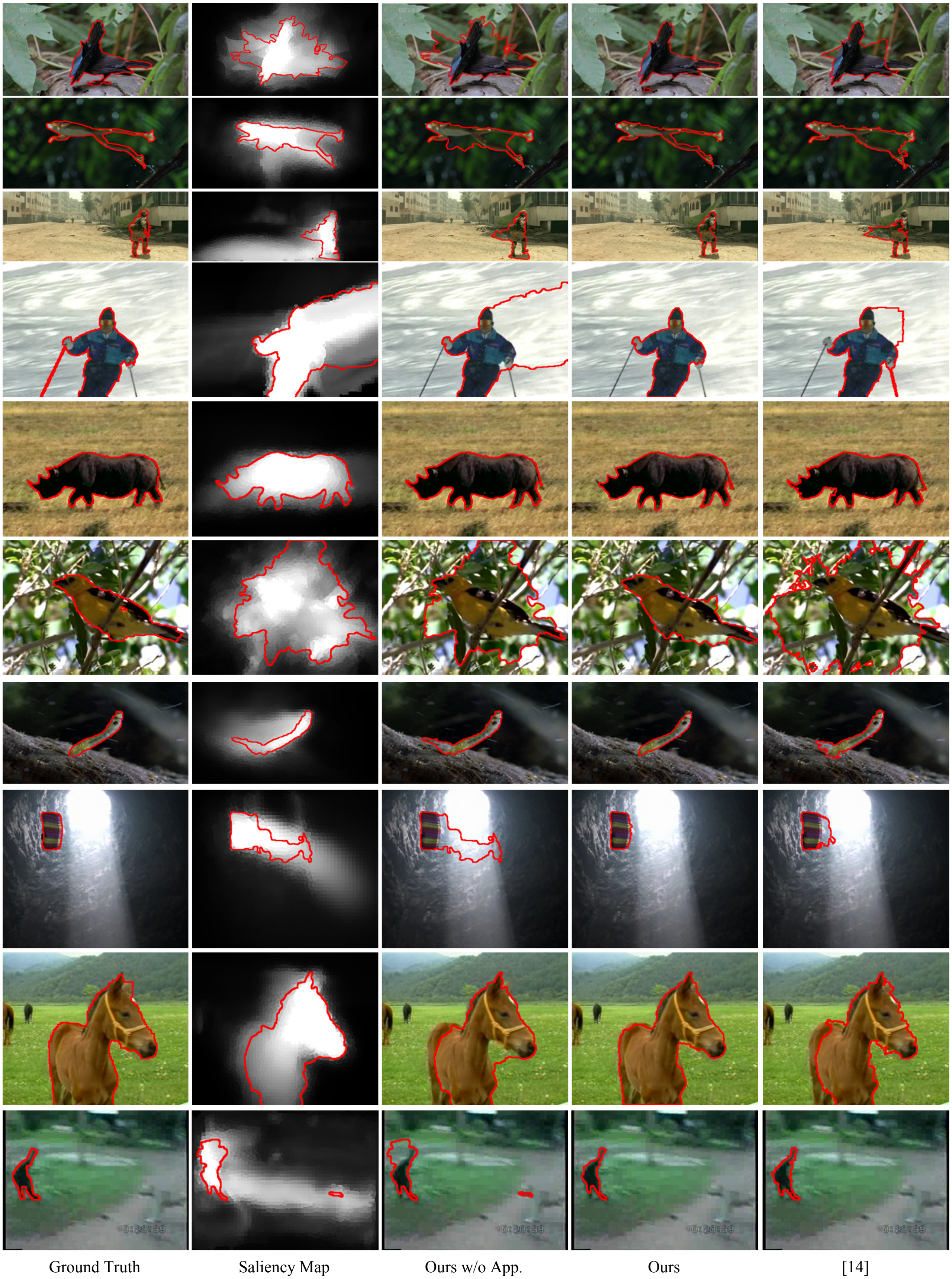

Fig. 5. Some qualitative results and comparisons.

TABLE I
COMPARISON RESULTS ON SEGTRACK V2 DATASET

video	video dimension	ours	ours w/o App.	[27]	[41]	[18]	[19]
bird_of_paradise	640 × 360 × 98	94.35%	76.12%	83.80%	-	92.20%	94.00%
birdfall2	259 × 327 × 30	66.23%	54.46%	59.00%	71.00%	49.00%	62.50%
frog	480 × 264 × 279	80.77%	47.16%	77.00%	74.00%	0.00%	65.80%
girl	400 × 320 × 21	81.78%	68.40%	73.00%	82.00%	87.70%	89.20%
monkey	480 × 270 × 31	68.48%	27.21%	65.00%	62.00%	79.00%	84.80%
monkeydog	320 × 240 × 71	78.25%	60.99%	79.00%	75.00%	-	58.80%
parachute	414 × 352 × 51	90.01%	60.36%	91.00%	94.00%	96.30%	93.40%
soldier	528 × 224 × 32	83.47%	64.78%	69.00%	60.00%	66.60%	83.80%
worm	480 × 364 × 243	81.66%	75.11%	74.00%	60.00%	84.4%	82.80%
average	-	80.55%	59.40%	74.53%	72.25%	69.40%	79.46%
runtime (seconds per frame)	-	6.84s	6.81s	15s	210s	>120s	240s

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

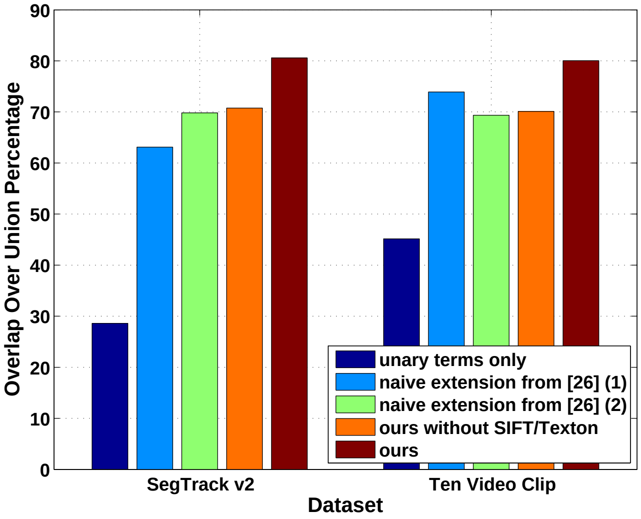

Fig. 6. Comparison with several baseline methods

reset to 7 by grid search to accommodate this change.

4) Our method without SIFT/Texton 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 overlap over union accuracy for both datasets are shown in Fig. 6. From the poor performance of baseline (1), it can be seen that the initial saliency estimation is far from a good segmentation. The comparison to baseline (4) shows the benefits of adding the dense SIFT and Texton feature by Cartesian Product. The comparison to baseline (2) and (3) and shows that it is not trivial to extend the method proposed in [33] 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 standard graph cut formulations, we evaluate the weight on the newly proposed appearance term in this section. In order to accomplish this, we conduct experiment to compare the segmentation accuracy by varying this weight and the results for both datasets are shown in Fig 7(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 starts 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 setting as described in Section IV-A and explicitly selecting the best weight for each individual video. The result is shown in Fig.7(c) and it can be seen that tunning the weight for each individual video can produce more accurate segmentations. However, the improvement is not significant due to the consistency 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. 8. The most common error is the inclusion of background region or exclusion of foreground region along the low contrast object boundaries such as the left leg of the frog in the first column of Fig. 8, the right arm of the monkey in the second column of Fig. 8 and the reflection of the monkey on the water surface in the third column of Fig. 8. This is the built-in difficulty of visual segmentation as it is very challenging for a computer to give very accurate boundary in these low contrast regions. As a human, we are able to find the correct boundary because we have prior knowledge about the object, for example we know that most monkeys have two arms and the object usually has a reflection on the water surface. The second common 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

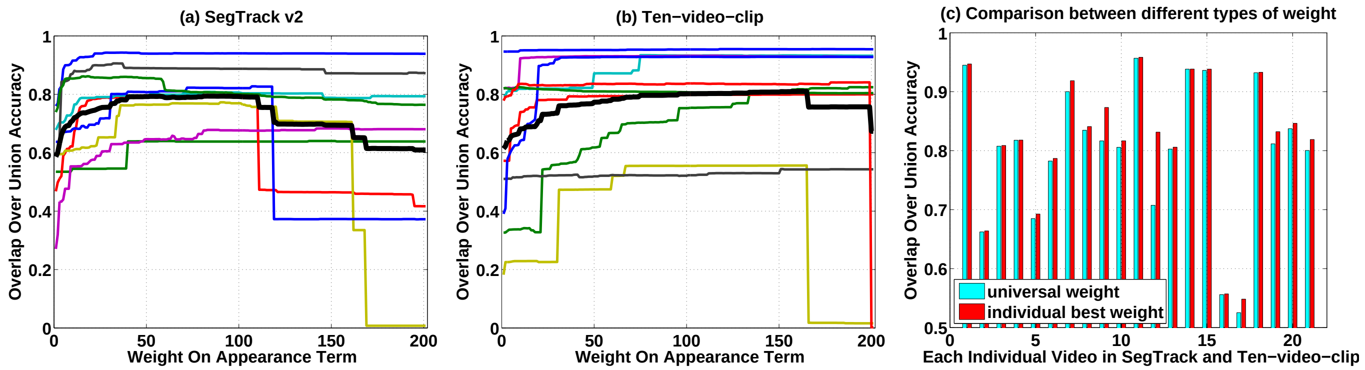

Fig. 7. Evaluation results regarding the weight on the appearance term. The first two curves show the segmentation accuracy of the SegTrack v2 and ten-video-clip dataset, respectively, by varying the weight from 0 to 200. The colorful thin lines indicate each individual video and the black thick lines indicate the average of each dataset. The right most bar plot shows the comparison of segmentation accuracy between using a universal weight setting as described in Section IV-A and selecting the best weight for each individual 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.

the second column of Fig. 8. These regions are labeled as foreground because they are fuzzed with high saliency value by the saliency warping/smoothing process along imperfect optical flows. The third common error is the misdetection of some thin structures attached to the main object such as the leg and foot of the bird in the last column of Fig. 8. These thin parts are either missed by the initial saliency estimation or smoothed away by the MRF inference. A common solution in the static image segmentation literature is to enforce higher order constraint of the MRF graph [17] and we leave this as our future work. In addition, the segmentation error in the fourth column of Fig. 8 is caused by corrupted saliency estimations. The saliency fails to highlight the lower part of the flower due to the cluttered background, *e.g.,* both the flower and the leaves are swaying in the wind. Moreover, there happen 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. Time Usage

As shown in Table I, our method is very efficient compared to the other approaches and the detailed time usage of the various stages is shown in Table III. The experiments are implemented in Matlab and conducted on a Dual-Core i5 PC with 8GB of RAM. Due to the good architecture of our method, we are able to parallelize many of the stages to achieve 6.84 seconds per frame[1]. Note that, the time statistics are based on the bird_of_paradise sequence in SegTrack v2 because it has the highest per-frame resolution. 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 main 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]. For the compared methods, [27] is slow because they adapt an iterative optimization process and need to run the EM algorithm for GMM estimation in each iteration. Also, [18], [41], [19] are significantly slow because they employ the more advanced but time consuming region proposals as the primitive input.

TABLE III
TIME USAGE OF THE VARIOUS COMPONENTS

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

V. CONCLUSION

In this paper, we have proposed 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 to many existing appearance models, our method is non-iterative and guarantees the global optimality in the optimization process. Experimental evaluations show that our method is faster than many of the alternatives and the segmentation accuracy is also better than or comparable to the state-of-the-art methods.

REFERENCES

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *PAMI*, 2012.

[1] We parallel the image saliency estimation, optical flow computation, feature extraction and superpixel segmentation and the two motion saliency estimations, *i.e.,* $6.84 = \max\{0.22, 1.25, 4.82, 0.80, 1.15, 0.15\} + \max\{1.08, 0.47\} + 0.34 + 0.58 + 0.01 = 4.82 + 1.08 + 0.34 + 0.58 + 0.01$

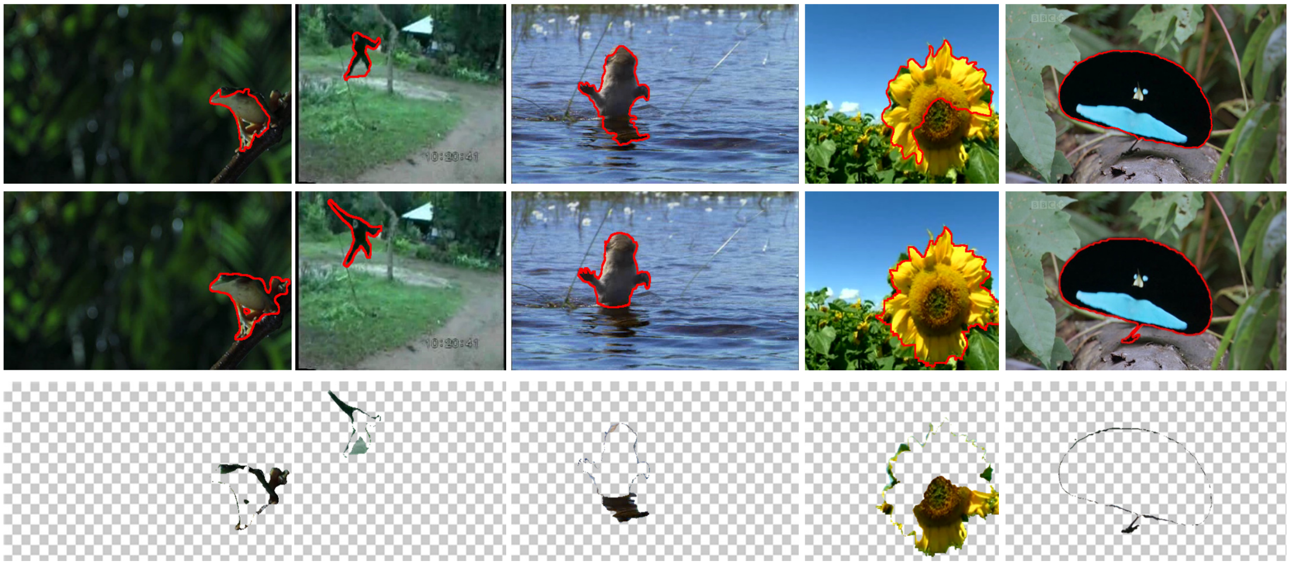

Fig. 8. Some typical segmentation errors. The first row is the segmentation result, the second row is the ground truth segmentation and the last row is the segmentation errors.

[2] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *T-PAMI*, 2012.

[3] V. Badrinarayanan, F. Galasso, and R. Cipolla. Label propagation in video sequences. In *CVPR*, 2010.

[4] X. Bai, J. Wang, D. Simons, and G. Sapiro. Video snapcut: robust video object cutout using localized classifiers. *ACM TOG*, 2009.

[5] D. Banica, A. Agape, A. Ion, and C. Sminchisescu. Video object segmentation by salient segment chain composition. In *ICCVW*, 2013.

[6] L. Bao, Q. Yang, and H. Jin. Fast edge-preserving patchmatch for large displacement optical flow. In *CVPR*, 2014.

[7] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *T-PAMI*, 2004.

[8] J. Carreira and C. Sminchisescu. Cpmc: Automatic object segmentation using constrained parametric min-cuts. *T-PAMI*, 2012.

[9] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu. Global contrast based salient region detection. *T-PAMI*, 2015.

[10] P. Dollár and C. L. Zitnick. Fast edge detection using structured forests. *PAMI*, 2015.

[11] I. Endres and D. Hoiem. Category independent object proposals. In *ECCV*, 2010.

[12] H. Fu, D. Xu, B. Zhang, and S. Lin. Object-based multiple foreground video co-segmentation. In *CVPR*, 2014.

[13] K. Fukuchi, K. Miyazato, A. Kimura, S. Takagi, and J. Yamato. Saliency-based video segmentation with graph cuts and sequentially updated priors. In *ICME*, 2009.

[14] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *CVPR*, 2010.

[15] S. D. Jain and K. Grauman. Supervoxel-consistent foreground propagation in video. In *ECCV*, 2014.

[16] C. Jung and C. Kim. A unified spectral-domain approach for saliency detection and its application to automatic object segmentation. *T-IP*, 2012.

[17] P. Kohli, P. H. Torr, et al. Robust higher order potentials for enforcing label consistency. *IJCV*, 2009.

[18] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *ICCV*, 2011.

[19] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, 2013.

[20] W.-T. Li, H.-S. Chang, K.-C. Lien, H.-T. Chang, and Y. Wang. Exploring visual and motion saliency for automatic video object extraction. *T-IP*, 2013.

[21] Y. Li, J. Sun, and H.-Y. Shum. Video object cut and paste. *ACM TOG*, 2005.

[22] C. Liu. Beyond pixels: Exploring new representations and applications for motion analysis. *MIT Doctoral Thesis*, 2009.

[23] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[24] Y. Luo, J. Yuan, P. Xue, and Q. Tian. Saliency density maximization for efficient visual objects discovery. *T-CSVT*, 2011.

[25] Y. Luo, G. Zhao, and J. Yuan. Thematic saliency detection using spatial-temporal context. In *ICCVW*, 2013.

[26] T. Ma and L. J. Latecki. Maximum weight cliques with mutex constraints for video object segmentation. In *CVPR*, 2012.

[27] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *ICCV*, 2013.

[28] B. L. Price, B. S. Morse, and S. Cohen. Livecut: Learning-based interactive video segmentation by evaluation of multiple propagated cues. In *ICCV*, 2009.

[29] P. Rantalankila, J. Kannala, and E. Rahtu. Generating object segmentation proposals using global and local search. In *CVPR*, 2014.

[30] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM TOG*, 2004.

[31] J. Shi and J. Malik. Normalized cuts and image segmentation. *T-PAMI*, 2000.

[32] J. Sun, H.-H. Lu, and X. Liu. Saliency region detection based on markov absorption probabilities. *T-IP*, 2015.

[33] M. Tang, L. Gorelick, O. Veksler, and Y. Boykov. Grabcut in one cut. In *ICCV*, 2013.

[34] J. Tighe and S. Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. In *ECCV*, 2010.

[35] D. Tsai, M. Flagg, A. Nakazawa, and J. M. Rehg. Motion coherent tracking using multi-label mrf optimization. *IJCV*, 2012.

[36] M. Van den Bergh, X. Boix, G. Roig, B. de Capitani, and L. Van Gool. Seeds: Superpixels extracted via energy-driven sampling. In *ECCV*, 2012.

[37] L. Wang, G. Hua, R. Sukthankar, J. Xue, and N. Zheng. Video object discovery and co-segmentation with extremely weak supervision. In *ECCV*, 2014.

[38] C. Xu and J. J. Corso. Evaluation of super-voxel methods for early video processing. In *CVPR*, 2012.

[39] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang. Saliency detection via graph-based manifold ranking. In *CVPR*, 2013.

[40] J. Yang, G. Zhao, J. Yuan, X. Shen, Z. Lin, B. Price, and J. Brandt. Discovering primary objects by saliency fusion and iterative appearance estimation. *Technical Report*, 2014.

[41] D. Zhang, O. Javed, and M. Shah. Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In *CVPR*, 2013.

[42] D. Zhang, O. Javed, and M. Shah. Video object co-segmentation by regulated maximum weight cliques. In *ECCV*, 2014.

[43] G. Zhao, J. Yuan, and G. Hua. Topical video object discovery from key frames by modeling word co-occurrence prior. In *CVPR*, 2013.

[44] L. Zheng, S. Wang, Z. Liu, and Q. Tian. Packing and padding: Coupled multi-index for accurate image retrieval. In *CVPR*, 2014.