Structural image and video understanding
Lou, Z.

Citation for published version (APA):
Lou, Z. (2016). Structural image and video understanding

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
4

Extracting Primary Objects by Video Co-Segmentation

4.1 Introduction

The amount of video data is explosively increasing. Every minute, more than 100 hours of video are uploaded to YouTube. Therefore, automatic video analysis is important for many applications such as object recognition, action recognition, video retrieval and editing.

In general, users are interested in primary objects such as a person, a car and so on [111]. Hence, instead of full video analysis, many video processing problems, such as content-based video search, are simplified by extracting primary objects. As shown in Figure 4.1, our aim is to extract the primary objects across videos automatically by video co-segmentation.

Algorithms for video object segmentation usually focus on single videos to extract a single primary object [85, 93, 143]. Exploiting optical flow [117] to capture motion patterns of the primary objects as well as other cues such as ‘objectness’ [2, 40], these methods obtain reasonable results considering a single primary object. However, in general, it is hard to determine the primary object since more than one object can be present. If a video contains many objects, small accidental disturbances may greatly influence the final result. Therefore, in this paper, we propose an algorithm considering a set of videos to extract the primary objects by applying video segmentation across videos.

Video co-segmentation is inspired by image co-segmentation [109, 129] in which the ambiguity of the definition of a primary object is studied. Without human annotation or other prior information, it is difficult to decide which object is the one to be extracted. However, with the approach by [109], one can learn the property of the primary object from other images to obtain a more meaningful segmentation result. Adopting this approach, [20, 110] consider the segmentation process of different videos simultaneously. One drawback is that these methods assume a binary foreground-background segmentation and therefore can only handle simple videos containing a single object. Furthermore, the initial object models are learned from segments obtained independently for each video. Hence, the final result may not converge to a single object. To handle multiple objects, [22] proposes a non-parametric algorithm to cluster pixels into different regions considering appearance and spatial-temporal information. As [22] clusters pix-
Figure 4.1: The aim is to extract primary objects by considering both the temporal coherence within a single video and the coherence across videos.

els by considering their similarities, the segmentation results are more likely to provide over-segmentations across videos.

In this paper, the object co-segmentation algorithm is based on the following contributions. First, to obtain an object model, all videos are considered in a probabilistic system exploiting the dependency between videos. Because a probabilistic model is used to learn the object segments, a semantic interpretation can be derived for the primary object. In previous methods, such as [22], a simple over-segmentation is generated in which the semantic interpretation of the obtained object segments is lost. Second, to extract the primary object, not only is the similarity between pixels considered, but other properties of the primary objects are used in the framework. The method can handle multiple videos and multiple objects. Moreover, we created a new dataset to evaluate our method and compare it to the state-of-the-art on video object co-segmentation. The code of the algorithm and the newly created dataset is publicly available [link]
4.2 Related Work

Our approach is related to three topics: image co-segmentation, video object segmentation and video co-segmentation.

**Image co-segmentation:** One of the most difficult problems of image segmentation is how to define an object. A simple solution is to use human labeling in the segmentation process. To mitigate user labeling, [109] proposes the concept of image co-segmentation to determine the primary object by considering information from two images. This concept is then extended to handle multiple images and multiple objects [73, 77] by considering spatial-temporal consistency within a video and appearance consistency across videos. The difference between image co-segmentation and our algorithm is that our approach considers spatial and temporal consistency of the primary objects within one video.

**Video segmentation:** Video segmentation is a prerequisite for many applications such as video summarization and video search. [60, 126] aim to parse a video into spatial-temporal consistent regions, called supervoxel. The number and size of the supervoxels are highly depending on the parameters of the algorithms. [85, 93, 143] extract the primary object from a single video automatically. These algorithms perform well on simple videos (mostly single objects in single videos). The difference between our method and these methods is that we learn the model of the primary object from multiple videos considering the properties of the primary object in a probabilistic system.

**Video co-segmentation:** Recently, video co-segmentation has been studied [20, 22, 110]. [20] extracts saliency regions from each video independently. The foreground and background models are learned based on the salient regions. Since the approach extracts salient regions independently from each video, consistent information between videos is ignored. The difference with our method is that we consider the consistency property among videos by means of a probabilistic approach to infer the primary object in each video. [22] considers video co-segmentation as a clustering problem. For different input videos, pixels are clustered into different groups based on appearance and spatial-temporal information. Although this algorithm can handle multiple objects in multiple videos, the segmentation result has no semantic meaning. In contrast, our approach extracts the ‘primary’ objects across videos.

4.3 Extracting Primary Objects by Video Co-Segmentation

As shown in Figure 4.2, the proposed framework has three components: 1) Firstly, object candidates are generated for each frame. 2) Then, a graphical model is built to select one primary object candidate for each frame. (3) Finally, foreground and background models are learned from the selected candidates and the corresponding background using Gaussian Mixture Models (GMM). A spatial-temporal graphical model is constructed and the final pixel-wise results are obtained for each video. In our framework, the first step generates object proposals [85, 93, 143]. The second step is the key part of our framework and it is the first time to exploit this type of graphical model to perform video co-segmentation. The last step uses GMM to refine the segmentation result [20, 85, 93, 143].
4. Extracting Primary Objects by Video Co-Segmentation

Figure 4.2: The framework of our approach consisting of three components: 1) Generating primary object candidates for each frame and for each video. 2) Selecting one segment from each frame by exploiting a graphical model. 3) Learning the foreground and background models and applying pixel-wise foreground-background segmentation based on a spatial-temporal graphical model.

For the ease of explanation, we assume that there are two videos on input. It is straightforward to extend the algorithm for multiple videos. To handle multiple objects, after extracting the primary object, we remove the primary object from the object candidates and repeat the algorithm to obtain the second-primary object.

4.3.1 Primary Objects

Depending on the application, the primary object could be anything like a person, animal, car and building. In table 4.1, the different properties used to describe primary objects are summarized. For example, [2, 40] introduces the property of objectness i.e. to what extent an image region looks like an object. Further, the relative motion of an object is used to describe the primary object. For example, [85, 93, 143] exploit relative motion between foreground and background to extract the primary object. [20] uses saliency
4.3. Extracting Primary Objects by Video Co-Segmentation

as an important cue to define a primary object. Repetitions within videos correspond to
the observation that primary objects should be observed as long as possible. Repetition
across videos is based on the assumption that primary objects are detected across videos.
Appearance saliency is also studied and exploited to determine the primary object [20].
In conclusion, to extract primary objects across videos, in this paper, we will use all the
features as shown in table 4.1.

Table 4.1: Features used to describe primary objects. The ‘OBJ’ denotes objectness. The
‘MOT’ represents motion. The ‘IAR’ represents intra video repetition which denotes the
repetition within one video. The ‘ITR’ represents the repetition across videos. The ‘SAL’
represents the appearance saliency.

<table>
<thead>
<tr>
<th></th>
<th>[101]</th>
<th>[85]</th>
<th>[93]</th>
<th>[143]</th>
<th>[20]</th>
<th>[22]</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJ</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>MOT</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>IAR</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>ITR</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>SAL</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
</tbody>
</table>

4.3.2 A New Graphical Model for Video Object Co-segmentation

Since our aim is to extract the primary object across videos, there is no prior information
about the object to be extracted. In fact, the primary object could be anything. Therefore,
we start by generating a large number of primary object candidates (segment/regions)
using the method proposed by [40]. The method provides an objectness score for each
image region. After generating a group of object regions for each frame, a graphical
model is built to select one candidate for each frame. As shown in Figure 4.3, each
shaded node is a group of (observed) object candidate regions generated for each frame.
Each blank node is a latent variable and it denotes the index of the image region to be
selected. There are two types of edges in the graphical model. One is called the intra-
video edge and the other is called the inter-video edge. The intra-video edge is used to
connect two adjacent frames within a video. In the graphical model, we add an intra-
video edge between every pair of adjacent frames. For each frame, we randomly select
a portion $\tau$ (in our experiments, we set $\tau=100\%$) of frames from other videos and add
an inter-video edge between this frame and all other selected frames. We compute the
labeling of each frame by minimizing the energy function defined as follows:

$$E(L) = -\log(\sum_{i=1}^{M} P_u(l_i)) + \sum_{(i,j) \in E_A} P_A(l_i, l_j)$$

$$+ \sum_{(i,j) \in E_I} P_I(l_i, l_j),$$

(4.1)

where $l_i$ is the latent variable for frame $i$. It is the index of one region to be selected for
frame $i$. There are $N$ region candidates, $l_i \in 1, 2, \ldots N$. $M$ is the total number of frames
Figure 4.3: The graphical model for detecting the primary object across videos. The shaded nodes are the observed variables representing the object candidates generated for each frame. The blank nodes are the latent variables representing the index of object candidates to be selected. The intra video edge is used to connect two adjacent frames while the inter video edge is used to connect two frames from different videos. The aim is to select one object candidate for each frame and enforcing the consistency between image frames.

in the video set. \( N \) is the number of region candidates for each frame. For each frame, we only keep the top ranked \( N \) proposals generated by the object proposal algorithm [40]. The unary term \( P_u(l_i) \) measures the potential of selecting the region \( l_i \) for frame \( i \). The pairwise terms \( P_A(l_i, l_j) \) and \( P_I(l_i, l_j) \) measure the intra-video and inter-video potential of selecting region \( l_i \) and \( l_j \) for frame \( i \) and \( j \) respectively. \( E_A \) are the edges between two adjacent frames within one video. \( E_I \) are the edges between two frames from two different videos.

**Unary potential** The unary potential consists of three parts: objectness, motion gradient and appearance saliency:

\[
P_u(l_i) = \exp(\alpha_1 + \alpha_2 \cdot f_o(l_i) + \alpha_3 \cdot f_m(l_i) + \alpha_4 \cdot f_s(l_i)).
\] (4.2)

Here, \( l_i \) is a candidate object region, \( f_o(l_i) \) is the objectness score for region \( l_i \). \( f_m(l_i) \) is the motion feature of region \( l_i \). \( f_m(l_i) \) measures the relative motion of the image region and its direct surrounding. The motion feature \( f_m(l_i) \) is obtained from [143] in which optical flow is computed for each frame using [117]. Then, the Frobenius norm of optical flow gradient is computed for each pixel of the image region. By collecting the Frobenius norm of optical flow gradients along the boundary of the object candidate, the
4.3. Extracting Primary Objects by Video Co-Segmentation

final motion feature is obtained \( f_m(l_i) \) as follows:

\[
f_m(l_i) = \frac{\sum_{p \in B(l_i)} \sqrt{u_x^2(p) + u_y^2(p) + v_x^2(p) + v_y^2(p)}}{|B(l_i)|}.
\]  

(4.3)

Here, \( u \) and \( v \) are the optical flow along the \( x \) and \( y \) axis respectively. \( B(l_i) \) is the boundary points of segment \( l_i \). \(|B(l_i)|\) is the total number of boundary points. For calculating the saliency score \( f_s(l_i) \), the method of [123] is used in which color boosting is obtained by [21]. In this way, a saliency map is computed for each region.

**Intra-video pairwise potential** The intra-video pairwise potential is defined between two adjacent frames. It measures the similarity of two object regions for two frames:

\[
P_A(l_i, l_j) = \exp(\beta_1 - \beta_2 \cdot C(l_i, l_j) - \beta_3 \cdot O(l_i, l_j)),
\]  

(4.4)

where \( l_i \) and \( l_j \) are two image regions from two adjacent frames. \( C(l_i, l_j) \) is the color histogram similarity between region \( l_i \) and \( l_j \):

\[
C(l_i, l_j) = |H_{l_i} - H_{l_j}|^2,
\]  

(4.5)

where \( H_{l_i} \) is the color histogram of the region \( l_i \). \( O(l_i, l_j) \) measure the shape, size and location consistency between two adjacent frames:

\[
O(l_i, l_j) = \left| \frac{l_i \cap W(l_j)}{l_i \cup W(l_j)} \right|^2,
\]  

(4.6)

where \( W(l_j) \) is obtained by warping region \( l_j \) to the frame of region \( l_i \) by optical flow measures. Intuitively, \( O(l_i, l_j) \) measures the overlap ratio of \( l_i \) and the warped region of \( l_j \). For example, if region \( l_i \) and \( l_j \) are from two different object, \( O(l_i, l_j) \) is very close to 0. In this case, the chance of choosing these two segments together is very low. However, if \( l_i \) and \( l_j \) are from the same object, \( O(l_i, l_j) \) is very close to 1. Hence, these two segments have a high probability to be chosen as an object.

**Inter-video pairwise potential** The inter-video edge potential is defined between two frames coming from two different videos. It measures the appearance similarity between two image regions. This term is crucial for video co-segmentation as it assumes that primary objects observed in different videos have similar appearance. The color histogram is used to measure the appearance differences:

\[
P_I(l_i, l_j) = \exp(\gamma_1 - \gamma_2 \cdot |H_{l_i} - H_{l_j}|^2),
\]  

(4.7)

where \( H_{l_i} \) is the color histogram of region \( l_i \). \( \gamma_1 \) and \( \gamma_2 \) are constants.

### 4.3.3 Decoding by Iterated Conditional Modes

The graphical model defined in the previous section is a conditional random field with cycles. It is an NP-hard problem to find the global optimal configuration of the graph. Therefore, we use iterated conditional modes (ICM) [9] to obtain an approximated solution. Suppose we have \( M \) frames in total. The observations are \( D = \{d_1, d_2...d_M\} \)
(each observation \(d_i\) is a set of regions for each frame). The corresponding labels are \(L = \{l_1, l_2, ..., l_M\}\). The aim is to find a configuration \(L\) to minimize the energy:

\[
L = \text{argmin}_L E(l_1, l_2, ..., l_M; D). \tag{4.8}
\]

When minimizing the energy function, we fix some labels and maximize the potential over the other labels as follows:

\[
l_i(k + 1) = \text{argmin}_{l_i} E(l_i; D, L - l_i). \tag{4.9}
\]

By updating the label for each frame until no label is changed, a minimum is obtained. However, the initial configuration of ICM is important and different initializations may result in different labels. To this end, we use two initial strategies. One is unary maximization in which the initial label for each frame has maximum unary potential. The other is random initialization in which a random initial number is adopted for each frame. By applying the algorithm with different initial labels, a number of configurations are obtained. We choose the one with minimum energy.

### 4.3.4 Spatial-temporal Graphical Model

In the previous section, the primary object is selected per frame. However, it is possible that for some frames, the selected segment is not the primary object. This is because the object candidate generation algorithm may fail to generate the primary object region.

To this end, a spatial-temporal graphical model is built to refine the candidate selection to avoid misdetections. In the graphical model, each pixel is a node. There are two types of edges in the graphical model: spatial and temporal edges. The spatial edges connect two neighboring pixels within one frame. The temporal edges are between two adjacent frames. For each pixel, temporal neighbor pixels in the next frame are obtained by optical flow. Then, the temporal edge is defined between this pixel and each of the corresponding temporal neighbor pixels.

Each node has two possible labels, foreground or background. The energy function contains three terms: a unary term, a spatial binary term and a temporal binary term:

\[
E = \sum_{i \in P} D_u(f_i) + \lambda_1 \cdot \sum_{(f_i, f_j) \in N_s} V_s(f_i, f_j) + \lambda_2 \cdot \sum_{(f_i, f_j) \in N_t} V_t(f_i, f_j). \tag{4.10}
\]

The unary term \(D_u(f_i)\) measures to what extent pixel \(i\) is similar to either foreground or background. Appearance and location are considered when computing the unary term. To obtain the appearance model, Gaussian Mixture Models (GMM) are learned for both the foreground and background of the primary object using the color values of the regions. Then the foreground probability for each pixel is generated. To obtain the location probability, the model projects each mask for the regions obtained to all other frames using optical flow. Each projected point generates a Gaussian distribution representing the foreground probability (as shown in Figure 4.4):

\[
D_u(i) = -\log(U_a(f_i) \cdot U_l(f_i)). \tag{4.11}
\]
4.4 Experiments

Figure 4.4: (a) The source image. (b) The foreground probability generated from the learned GMM. (c) The location probability. (d) The unary term of each node in the graphical model. (e) The boundary strength $p_b$ [4]. (f) The gradient of optical flow.

The spatial binary term $V_s$ corresponds to the cost of assigning different labels to a pair of spatially neighboring pixels. Two features are considered to measure the cost: the boundary strength $p_b$ [4], and the gradient of optical flow:

$$V_s(i, j) = \left[ f_i \neq f_j \right] \cdot (\mu_1 - \mu_2 \cdot M(f_i, f_j) - \mu_3 \cdot G(f_i, f_j)). \quad (4.12)$$

The temporal binary term $V_t$ corresponds to the cost of assigning the same labels or different labels to a pair of temporally neighboring pixels. The opponent RGB is used to compute this term:

$$V_t(i, j) = \left[ f_i \neq f_j \right] \cdot (\eta_1 - \eta_2 \cdot |H_{f_i} - H_{f_j}|^2) \quad (4.13)$$

To obtain the optimal solution efficiently, we use the ICM [9] to minimize the energy function defined in eq.(4.9) and obtain the final segmentation result. We also consider iterating between appearance modeling and segmentation. To reduce the computational complexity, we retrain the appearance models only once.

4.4 Experiments

In this section, we evaluate our algorithm on the dataset MOViCS [22] and our new Sports dataset. The influence of different parameter settings are evaluated. To handle multiple objects, the number of objects are given in this experiment.
4. Extracting Primary Objects by Video Co-Segmentation

4.4.1 Implementation

For some videos, the object might be out of the sight for a number of frames. To this end, an extra ‘non’ label is added to each node in the graph described in section 4.3.2. This ‘non’ label represents that no object candidate is selected for that node. The unary and binary potentials related to ‘non’ labels correspond to the average potential of all the unary and binary potentials in the graphical model. If there is no object present in a frame, the algorithm will not select any object region for that frame. Another approach is to group all the selected segments into N (N=3 in our experiments) groups by running the K-means algorithm multiple times with random initialization and finally choose the solution with the minimal cost. For each group, a score is generated by aggregating the unary potential in eq.(4.2) of each region in the group. The group with the maximum score is taken for learning the foreground models.

We use the algorithm of [40] to obtain the object candidates for each frame. To describe each object region, a histogram in opponent RGB space is computed with 39 bins (13 bins for each channel). To obtain the motion feature, we use the algorithm of [117] with the option ‘classic+nl-fast’ to generate the optical flow. When computing the motion feature given by eq.(4.3), the boundary is dilated by 3 pixels. This is to compensate for the potential deviation of the optical flow. We train the GMM described in section (4.3.4) with k=5 components.

4.4.2 Comparison on the Multi-Object Video Co-Segmentation (MOViCS) Dataset

[22] proposes a dataset, called MOViCS, for evaluating video co-segmentation algorithms. This dataset contains four sets of videos, 11 videos in total, and each set has 2-4 videos. For each set, there are one or two common objects. The pixel-level ground-truth segmentation is available for every object. To handle the case of multiple objects, after obtaining the primary object, the obtained primary object is removed from the original video. Specifically, we remove those object proposals which have large overlap with the obtained primary object. Then, the algorithm is executed for a second time. In this way, we can extract the primary object, second-primary object and so on. To evaluate our algorithm, we follow the setup of [22] and compare our algorithm with three other state-of-the-art algorithms [22, 73, 101]. The intersection-over-union metric defined as:

$$A(R, G) = \left| \frac{R \cap G}{R \cup G} \right|$$  \hspace{1cm} (4.14)

is used to measure the accuracy. Here, $R$ is the result mask and $G$ is the ground truth mask. For the results of [22, 73, 101], the segments which have the maximum overlap with the ground-truth are selected to be the segment of the primary object.

As shown in table 4.2, the proposed algorithm outperforms all other algorithms on three videos set. The average accuracy of the proposed algorithm is 52.0%. It outperforms the video co-segmentation algorithm [22] by 1.5%, image co-segmentation algorithm [73] by 34.8% and video segmentation algorithm [101] by 25.4%. The performance of our method is the highest on three sets of the videos. Although the ‘Tiger’ set also has large motion (please refer to the supplementary videos), the appearance of different
4.4. Experiments

Table 4.2: Segmentation accuracy. We compare the proposed algorithm to the state-of-the-art algorithms (Image Co-segmentation [73], Video Segmentation [101], Video Co-segmentation [22] and [22] with motion features from [117] on the data set MOViCS.

<table>
<thead>
<tr>
<th>Video Set</th>
<th>[73]</th>
<th>[101]</th>
<th>[22]</th>
<th>[22]+[117]</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChickenTurtle</td>
<td>7.9</td>
<td>13.1</td>
<td>64.8</td>
<td>63.8</td>
<td>67.7</td>
</tr>
<tr>
<td>LionZebra</td>
<td>23.8</td>
<td>20.9</td>
<td>47.7</td>
<td>54.1</td>
<td>54.1</td>
</tr>
<tr>
<td>ElephantGiraffe</td>
<td>6.5</td>
<td>26.1</td>
<td>52.3</td>
<td>54.1</td>
<td>56.1</td>
</tr>
<tr>
<td>Tiger</td>
<td>30.64</td>
<td>46.4</td>
<td>30.1</td>
<td>30.0</td>
<td>30.0</td>
</tr>
<tr>
<td>Average</td>
<td>17.2</td>
<td>26.6</td>
<td>48.7</td>
<td>50.5</td>
<td>52.0</td>
</tr>
</tbody>
</table>

videos is quite different and the foreground and background is very similar. However, [101] considers only motion which ignore the influence of background and take the advantage in this specific video set.

Since image co-segmentation [73] only exploits the similarity of pixels in different frames, all the video-based segmentation algorithms which consider the spatial-temporal consistent within each video outperform the image co-segmentation algorithm. For [101], points are tracked and then clustered into a spatial-temporal region based on the motion properties of each pixel. Hence, this algorithm works when there is a large motion of the object. As shown in Figure 4.5, when the motion is not very significant (the giraffe in the second row has no motion) or the motion is not very accurate (the chicken in the last row is blurred which results in inaccurate motion), the video segmentation algorithm [101] can not generate meaningful objects. However, the proposed algorithm extracts the primary object across videos. If there is no motion, other videos help to extract the primary object in each video. Besides higher accuracy, another advantage of the proposed algorithm is that the primary object has a semantic meaning. The result of [22] is over-segmented and the results are highly dependent on the parameters. Figure 4.5 shows a number of visual results for the different segmentation algorithms. The objects colorized by red are recognized as the primary object and the objects colorized by green is recognized as the second primary object. Note that, for the ‘ElephantGiraffe’ set, the elephant is recognized as the primary object since the elephant moves fast and occurs in the video for much larger time while the giraffe in the second row has no motion. Occlusion may influence the performance of the segmentation. If an object is occluded by other objects, there will be a temporal discontinuity. However, the effect of occlusion is reduced by two other properties of the proposed model. Firstly, an extra label ‘non’ is explicitly added to denote that no object candidates are selected for that frame. Secondly, since each frame is connected to all frames in the other videos, the primary object can still provide a high score because of the inter-video edges. In the Elephant Giraffe video, the elephant is occluded by the giraffe in the beginning of the video. In the Lion Zebra video, the lion is occluded by the zebra in a number of frames. In both cases, our algorithm performs well.

In the proposed algorithm, ICM is used for decoding. In general, graph cuts or loopy belief propagation may outperform ICM. However, in the proposed model, each frame is connected to hundreds of frames. Therefore, the edges in the model is very dense. For this kind of graphs (i.e. many edges), ICM is better suited and hence may outperform
4. Extracting Primary Objects by Video Co-Segmentation

graph cuts or belief propagation. For comparison, Loopy Belief Propagation (LBP) and Graph-Cut (Alpha-Expansion) are evaluated. In Table 4.3, the results using different decoding methods are shown.

<table>
<thead>
<tr>
<th>Video Set</th>
<th>ICM</th>
<th>Graph-Cut</th>
<th>LBP</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChickenTurtle</td>
<td>67.7%</td>
<td>51.0%</td>
<td>61.5%</td>
</tr>
<tr>
<td>LionZebra</td>
<td>54.1%</td>
<td>56.1%</td>
<td>56.3%</td>
</tr>
<tr>
<td>ElephantGiraffe</td>
<td>56.1%</td>
<td>45.8%</td>
<td>44.8%</td>
</tr>
<tr>
<td>Tiger</td>
<td>30.0%</td>
<td>33.1%</td>
<td>38.3%</td>
</tr>
<tr>
<td>Average</td>
<td>52.0%</td>
<td>46.5%</td>
<td>50.2%</td>
</tr>
</tbody>
</table>

We run the algorithms on a computer containing an Intel Xeon CPU E31270 3.40GHz with 16G RAM. In Table 4.4, the computational time for each video set is given.

Table 4.4: The time consumption of different video set (Unit: seconds).

<table>
<thead>
<tr>
<th>Video Set</th>
<th>Total time (s)</th>
<th>Time per frame (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChickenTurtle</td>
<td>6500</td>
<td>57.7</td>
</tr>
<tr>
<td>LionZebra</td>
<td>7438</td>
<td>71.5</td>
</tr>
<tr>
<td>ElephantGiraffe</td>
<td>15691</td>
<td>64.1</td>
</tr>
<tr>
<td>Tiger</td>
<td>2516</td>
<td>41.2</td>
</tr>
</tbody>
</table>

4.4.3 The Influence of Different Parameter Settings

In automatic video object segmentation, the parameter settings may influence the performance of the different approaches. For the algorithms of [143] and [22], parameters are tuned manually. However, using their default parameters, the algorithm may fail for new videos. Therefore, in this section, the influence is analyzed for different parameter settings of our algorithm. The influence is studied for the three unary features: objectness, motion and appearance saliency. We exclude the spatial-temporal refinement step as it may influence the final result. Hence, only the performance of the primary object selecting step is reported in Figure 4.6. It can be derived that, for all features, an increase in parameter values results in a decrease in accuracy for most videos. For the feature ‘objectness’ and ‘saliency’, the method is not very sensitive to small changes. If we keep the parameters around 0.4 to 1, the performance remains the same. But for the ‘motion’ feature, it influences the performance more severely. In conclusion, the parameter settings of ‘objectness’ and ‘saliency’ are stable and can be fixed (around 0.4 to 1). The parameter setting of ‘motion’ requires careful adjustment.

4.4.4 Evaluation on New Sports Dataset

In the dataset MOViCS [22], each object has a single color or texture. However, for many real videos, an object is composed of different parts with a different color/texture. To this
4.5 Conclusion

Table 4.5: Segmentation result on the new Sports dataset. We compare our algorithm with both the state-of-the-art Video Co-segmentation [22] and single video segmentation [143] algorithm on this dataset. For [143], it cannot generate good result after the MRF step (their default parameter may not suitable for this dataset), we also report the result of the selecting step.

<table>
<thead>
<tr>
<th>Sets</th>
<th>ViCoseg[22]</th>
<th>ViSeg-Select[143]</th>
<th>ViSeg-Final[143]</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bolt</td>
<td>56.7%</td>
<td>47.1%</td>
<td>29.6%</td>
<td>60.2%</td>
</tr>
<tr>
<td>Bryant</td>
<td>53.2%</td>
<td>53.5%</td>
<td>10.2%</td>
<td>52.6%</td>
</tr>
<tr>
<td>Woods</td>
<td>45.2%</td>
<td>59.9%</td>
<td>27.4%</td>
<td>71.9%</td>
</tr>
<tr>
<td>Average</td>
<td>51.7%</td>
<td>53.5%</td>
<td>22.4%</td>
<td>61.3%</td>
</tr>
</tbody>
</table>

end, we have created a new dataset consisting of 3 sets, 6 videos in total. The videos are taken from different sports (basketball, golf and athletes). The videos are obtained by searching keywords ‘Kobe highlights’, ‘Woods highlights’ and ‘Bolt highlights’ on Youtube. We manually labeled the ground-truth and report the performance of different algorithms in table 4.5.

A number of visual results are shown in Figure 4.7. It can be derived that for the ‘basketball’ and ‘golf’ sets, the method by [143] extracts a number of non-primary objects (see the first row) or provides only a part of the primary object (see the fifth row). However, since we consider the primary object across videos, even if there is more than one object, our method is able to extract the primary one. The method of [22] clusters pixels by measuring their similarity. When different parts of the object have different appearance, this approach is not able to generate the object layouts.

As shown in table 4.5, the proposed algorithm outperforms the other methods on two sets of videos. The average accuracy of our method is nearly 9.6% better than video co-segmentation algorithm [22] and 7.8% better than single video object segmentation algorithm [143].

4.5 Conclusion

In this paper, we have proposed a graphical model to extract the primary objects across videos. A novel probabilistic graphical model is introduced to select the primary object from each frame. Exploiting this graphic model, we considered both the ‘objectness of image regions and the dependency between frames within one video and across videos in a whole framework. The method can handle multiple videos and multiple objects.

We have created a new dataset to evaluate our method and compare it to the state-of-the-art on video object co-segmentation. Our algorithm obtains state-of-the-art results on the Multi-Object Video Co-Segmentation (MOViCS) dataset and the Sports dataset outperforming existing methods.
Figure 4.5: Comparison results for different state-of-the-art methods. From left to right: the original video, the ground-truth, the results of image co-segmentation[73], the results of video segmentation [101], the results of video co-segmentation of [22] and our method. For the result [22, 73] and ours, the same color represent the same class of objects in each video set.
4.5. Conclusion

Figure 4.6: Evaluation of the influence of different parameter settings. In each Figure, all parameters are fixed except for the parameter under consideration.

Figure 4.7: Results obtained for the new Sports dataset. Since the method of [143] can not generate good results with the default parameters, the selected regions which are generated by the key-segments selecting step in their method is shown. For the other two methods, the final results are shown.