Consistent Foreground Co-segmentation

Jiaming Guo[1], Loong-Fah Cheong[1], Robby T. Tan[3] and Steven Zhiying Zhou[1,2]

[1]Department of ECE, National University of Singapore, Singapore
[2]National University of Singapore Research Institute, Suzhou, China
[3]SIM University, Singapore

Abstract. When the foreground objects have variegated appearance and/or manifest articulated motion, not to mention the momentary occlusions by other unintended objects, a segmentation method based on single video and a bottom-up approach is often insufficient for their extraction. In this paper, we present a video co-segmentation method to address the aforementioned challenges. Departing from the objectness attributes and motion coherence used by traditional figure-ground separation methods, we place central importance in the role of "common fate", that is, the different parts of the foreground should persist together in all the videos. To accomplish this idea, we first extract seed superpixels by a motion-based figure/ground segmentation method. We then formulate a set of linkage constraints between these superpixels based on whether they exhibit the characteristics of common fate or not. An iterative constrained clustering algorithm is then proposed to trim away the incorrect and accidental linkage relationships. The clustering algorithm also performs automatic model selection to estimate the number of individual objects in the foreground (e.g. male and female birds in courtship), while at the same time binding the parts of a variegated object together in a unified whole. Finally, a multiclass labeling Markov randome field is used to obtain a refined segmentation result. Our experimental results on two datasets show that our method successfully addresses the challenges in the extraction of complex foreground and outperforms the state-of-the-art video segmentation and co-segmentation methods.

1 Introduction

Imagine how, starting with a lack of models for most categories of objects, a developing young infant, say 7-8 month old, can come to acquire the faculty of segmenting the world into objects. It is believed that young infants gradually perceive individual objects as unified, bounded, and persisting by repeated observations from different perspectives and how objects interact with others [1]. However, the computational process underpinning this developmental process is not well-explored. Imagine another (common) scenario where we are given multiple videos with the same tag, but no further information is provided; how can we automatically augment the tag with more fine-grained information such as the segmentation of the tagged object [2]? These two scenarios provide the motivation for our work.

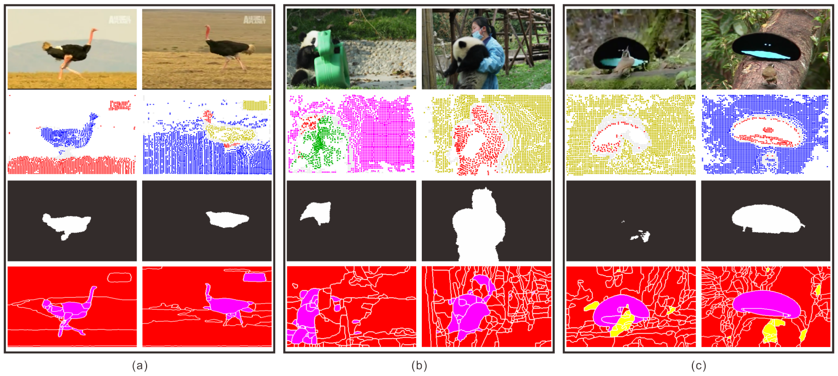


Fig. 1. Challenges of video foreground co-segmentation: variegated objects (such as the ostrich and the panda's variegated black and white appearance), objects hardly separable from the background (such as the inconspicuous female Bird of Paradise in (c)), and motion ambiguities caused by articulated motions of many animals, and extraneous objects moving together momentarily by chance (e.g. the green toy horse in (b)). **First row**: Original videos. **Second row**: Video segmentation results from [10]. **Third row**: The selected object proposals of [8]. **Fourth row**: Results of the proposed video foreground co-segmentation method.

Our work is akin to the traditional figure-ground separation albeit in a multiple video setting. We prefer calling it foreground separation rather than figure-ground separation in such multiple video setting, as not necessarily all the figures in the individual videos are of interest — some figural objects are only present fleetingly and/or coincidentally. Despite some such subtle differences, our problem has many similarities with the traditional figure-ground separation works. Of course, figure-ground separation has been a longstanding important problem. Despite many attempts made over decades [3–9], the problem remains difficult or even ill-defined. In those methods based on a single image, classical mid-level visual cues to figure/ground assignment such as convexity and parallelism are used [3–6]. However, most proposed representations are still too local and bottom-up to handle the complex variability in natural images. They were usually demonstrated solely on line images, with a few exceptions [4, 5].

The reason why figural assignment is hard is because it is not a purely bottom-up phenomenon [11]. Top-down cues such as familiar shape contours play a role [12], especially in natural scenes where many objects may not have convex shape or have holes. Moreover, the figure itself may contain multiple objects, which may be spatially separated with each other so that many of the figure-ground segmentation methods may fail to extract the whole figure due to their continuity assumption about the figure.

When we are viewing a dynamic scene, motion cue provides strong information about figural assignment. Despite the utility of motion cues, not many methods exploit motion for figure extraction. Recently, video segmentation methods [10, 13, 9] divide the video into motion layers, though the focus of these methods has been not so much on figure-ground separation. For simple scenes (e.g. near planar) or object motions (e.g. rigid), these approaches of course *also* yield figure and ground as two layers, but for more complex scenes and object motions, this simple strategy would fail. The natural world, unfortunately, abounds with such motions, such as the slithering motion of snake, the articulated motion of ostrich in Fig. 1(a), and indeed, almost all animal motions. The video segmentation approach is also plagued by the practical difficulties of obtaining accurate optical flow. Fig. 1 illustrates two of these difficulties. In the third row of Fig. 1(a), the elongated head and neck of the ostrich are poorly delineated because of the well-known short-boundary bias of standard pairwise MRF model and the failure of the objectness measure [14, 15] to cover the whole figure; in Fig. 1(c), the smaller female Bird of Paradise in the near ground fails to be separated from the background due to the paucity of textural details in the female bird.

From the above brief review of the figure-ground separation problem, we can make the following observations. The image-based methods are often plagued by over-segmentation, due to the variegated appearance of many objects and the non-convex shapes of many real-world objects. While the image-based methods can use motion cues to bind object segments together, they often over-rely on motion coherence which limits its applicability for natural motions. The use of motion cue also does not guarantee accurate figure outline due to the practical difficulty of estimating optical flow (See the second row of Fig. 1).

In our problem setting, the use of motion (or form for that matter) brings another complication: How to determine whether a group of segments (coherent in motion or form) are from the same object, but not from different objects moving together momentarily? One example is shown in Fig. 1(b) where a panda is playing with a toy horse. In other videos, there might be multiple moving objects. Some might be only momentarily present, but some might be interacting with one another on a prolonged basis, for instance, the two Birds of Paradise in courtship ritual in Fig. 1(c). In the former example, imagine we are trying to segment all the pandas in a group of videos bearing the tag “panda”. Then, clearly we are not interested in the toy horse. In the latter example, there might be strong reasons to regard the multiple objects as a single foreground entity.

In solving our problem of video foreground separation, we need to handle the aforementioned difficulties faced by the image-based approach as well as those using dynamic cues. Our definition of foreground is much more generic than those used for figure-ground separation; we eschew assumptions used by the preceding approaches, such as those based on objectness and motion coherence. As we have at our disposal multiple video sequences, with the foreground of interest appearing in all of them, the foreground is simply an object that is recurring in all the videos, moving differently from the background but having certain permanence quality about it. Operationally, this permanence quality is checked by requiring the different parts of the foreground should persist together in all the videos. In other words, the goodness of the foreground is based upon “common fate”, which we believe is a much more generic assumption than those used for figure-ground separation. By observing the appearance under different environment, we will be able to tease out the stable from the accidental, not getting entangled in possibly spurious correlations of features.

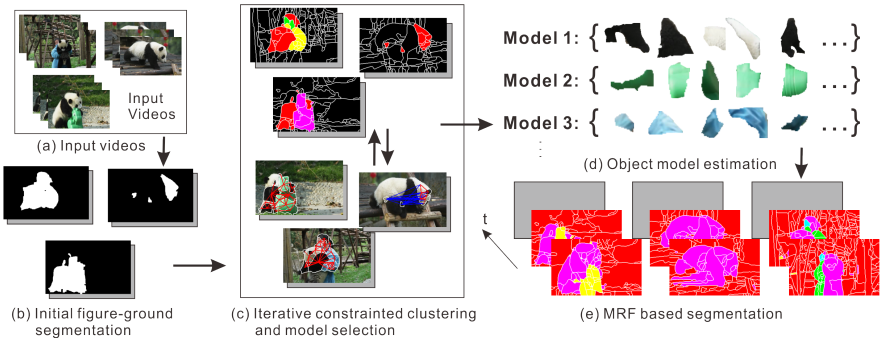

Input Videos
(a) Input videos
(b) Initial figure-ground segmentation
(c) Iterative constrainted clustering and model selection
Model 1: { ...}
Model 2: { ...}
Model 3: { ...}
(d) Object model estimation
t
(e) MRF based segmentation

Fig. 2. Algorithm overview with steps (a) to (e).

Fig. 2 shows an overview of the proposed method. It first performs an initial motion based figure-ground segmentation within each video to get seed superpixels for foreground and background. We also generate initial pair-wise to-link and not-to-link constraints between these superpixels based on whether they manifest the characteristics of common fate or not. Using these seed superpixels and their pair-wise constraints, we propose an iterative constrained clustering algorithm, in which the grouping together of articulated or variegated objects is promoted by retaining and making use of the correct and common constraints, whereas the removal of spurious connections is cast as discovering and pruning of violated constraints. We also need to perform model selection in the constrained clustering step because we want to allow for multiple objects in the foreground. Finally, we perform multiple and multiclass labeling Markov random fields (M-RFs) to obtain the final refined co-segmentation result.

We test our method on a newly created dataset, CFViSC as well as on the MOViSC dataset from [16]. The videos of CFViSC highlight the aforementioned video foreground segmentation challenges. Our experiments in Sect. 3 show that our method successfully addresses these challenges and outperforms the state-of-the-art video segmentation [10, 13, 8] and co-segmentation [16] methods in term of foreground segmentation accuracy.

1.1 Related Works

Video Segmentation: Video segmentation methods such as [10, 13] make use of dense trajectories and the associated motion cues for grouping. Due to the lack of explicit notion of how the figure looks like, they simply assume that the figure is the content moving in the scene. Clearly, this is not fine-grained enough in many cases where some extraneous objects of no interest are also moving or momentarily interacting with the figure. Another limitation of these methods arises when there are objects with articulated motions. In this case, relying on the pair-wise motion distance is likely to result in over-segmentation.

Some other methods make use of dense optical flow between two frames for figure-ground segmentation [17, 18]. They are also easily plagued by the practical difficulties of obtaining accurate optical flow. The work of [9] aims to address this issue by simultaneously estimating accurate flow and solving for a figure-ground segmentation that yields good flow estimates. While it is able to recover

Method	MS	MF	CM	Hetero-FG
Ours	Y	Y	Y	Y
ddCRP [16]	Y	Y	Y	N
ObMiC [21]	N	Y	N	Y
SC&QPBO [22]	N	N	N	Y
DC-M [23]	N	Y	N	N
MFC [24]	N	Y	Y	Y
OC [20]	N	N	N	Y

Table 1. Comparison of our algorithm with previous video and image co-segmentation methods (top and bottom halves respectively). MS: whether an algorithm can perform model selection. MF: whether an algorithm is designed for multiple figure object segmentation. CM: whether an algorithm can deal with the content misalignment issues (see text for discussion). Hetero-FG: whether an algorithm can identify a heterogeneous object as a single object. Y and N represent yes and no respectively

fine structures, it still faces the limitation of a two-layer segmentation and would suffer from the various ambiguity problems mentioned above.

Recently, several video segmentation methods built upon object proposals [14, 15] are proposed to detect the primary object in videos [7, 8]. When faced with the scenario in Fig. 1(b), they are still likely to suffer from the aforementioned issue as they are unable to determine whether there is an object with variegated appearance or there are multiple objects moving together. Other modes of failure include: the employed object proposal method may fail to generate adequately good proposals to correctly cover the whole figure. Even when there exist good object proposals, the segmentation algorithm may fail to identify them and select the bad ones. This is likely to happen especially when the object has non-compact shape. For instance, in Fig. 1(a), due to the variegated appearance and its articulated motion, the selected object proposals by [8] does not cover the neck and the feet of the ostrich (third row, Fig. 1(a)).

Co-segmentation: The problem of object co-segmentation is first addressed by [19] on an image pair. The usage of object proposals has also been introduced to co-segmentation [20, 21]. They share with other object-based approaches the same limitation mentioned in the preceding paragraph.

There are other co-segmentation works that segment objects using videos [22, 16]. The former [22] formulates subspace clustering for video co-segmentation which jointly utilizes appearance feature across multiple videos and motion features within each video to segment the foreground of interest. The assumption that the motion of each object forms a low-rank subspace makes this work incapable of handling objects with articulated motion. While it can treat multiple objects as foreground, it cannot provide further segmentation into the individual foreground objects. The latter [16] formulates a distant-dependent Chinese Restaurant Process across multiple videos based on motion cues and appearance cues, but the co-segmentation results are not organized into foreground and background explicitly. It also suffers from severe over-segmentation when dealing with complex scenes with a lot of clutter.

Table 1 compares our method with the previous video and image object co-segmentation methods. As explained above, our method can handle foreground with variegated appearance and non-compact shape, foreground comprising multiple objects (with the number of objects unknown), and finally, can remove extraneous or spurious objects momentarily present in the scene or interacting with the foreground. Note from Table 1 that [16] and [24] are also able to handle extraneous objects that are only present in some of the scenes. They termed this kind of images or videos as exhibiting content misalignment. One big difference is that they choose to retain these extraneous objects in the foreground. In principle, both these two and our methods utilize the information to discard or retain these extraneous objects, depending on the needs of the applications.

2 Proposed Method

Given a set of N videos $\mathcal{V} = \{V_1, V_2, ..., V_N\}$, we first run the motion-aware superpixel segmentation of [25] for each frame within each video, and then represent each video as a collection of superpixels, i.e., $\mathcal{S} = \{S_1, S_2, ..., S_N\}$, where S_i denotes the superpixel collection of V_i. Our video co-segmentation method presented in this section is based on these superpixels as input.

2.1 Discovering Seed Superpixels and Initial Pairwise Constraints

The objective in this step is to perform a rudimentary foreground-background segmentation in each video to obtain a set of seed superpixels and some initial pairwise constraints among these selected superpixels. In this rudimentary foreground-background segmentation, often only fragments of the foreground are selected, together with extraneous background or other undesirable objects. Thus, further processing of foreground-background separation will be needed.

To extract the seed superpixels, we adopt the latest technique in computing the motion saliency map [18] and the inside-outside map [26]. The motion saliency measure of [18] exploits the center-surround difference on optical flow field to separate the foreground. It is relatively robust to any complex intra-object motion differences that could arise from self-occlusion or articulated motion. For instance, it allows the head and neck of the ostrich in Fig. 1(a) to have different motion from the body, as long as the contrast with the background is large enough. The drawback is that it depends on sufficient motion contrast (see the missing arm in the second column, second row of Fig. 3), and could be sensitive to spurious motion contrast due to depth discontinuities in the background.

The inside-outside map of [26] is based on first detecting motion boundaries, and then based on these incomplete boundaries, computes the inside-outside map via a point-in-polygon rule. Specifically, any ray starting from a point inside the polygon (or any closed curve) should intersect the boundary of the polygon an odd number of times. According to our observation, this inside-outside measure significantly outperforms the motion saliency measure when the foreground has a small motion contrast against the background. However, it is erroneous when the foreground object possesses large intra-object motion differences, since in this case, the differences could raise too many edges in the interior of the foreground, violating the basic premise of the point-in-polygon rule.

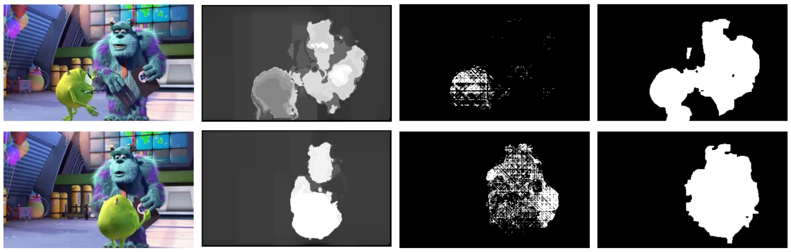


Fig. 3. First column: Two original frames. **Second column**: motion saliency measure. **Third column**: inside-outside measure, with intensity indicating degree of insideness. **Fourth column**: extracted patches by combining motion saliency and inside-outside measure.

Fig. 3 shows both the motion saliency map and the inside-outside map of two frames of a video, where their aforementioned pros and cons are well-illustrated in the second and third columns. We found that the two measures can actually complement each other to resolve their drawbacks. Thus, we combine them to extract those seed superpixels s that are likely to cover the foreground region:

$$\mathcal{S}^{\mathrm{F}} = \{s \,|\, \overline{\mathrm{sal}}(s) > \alpha \;\text{ or }\; \overline{\mathrm{in}}(s) > \beta\}, \tag{1}$$

where $\mathcal{S}^{\mathrm{F}}$ denotes the collection of seed superpixels (F stands for figure); $\overline{\mathrm{sal}}(\cdot)$ and $\overline{\mathrm{in}}(\cdot)$ represent the average motion saliency and the inside points ratio of a superpixel respectively; α and β are the thresholds. The fourth column of Fig. 3 shows initial foreground-background separation results. Despite the relatively good result of the foreground-background segmentation for this example, there are plenty of other examples where the initial segmentation is inadequate, for instance, the panda shown in Fig 2.

After the rudimentary foreground-background segmentation, our next aim is to generate the pair-wise constraints among the extracted seed superpixels of each input video. These constraints will eventually guide the formation of the correct foreground model in a constrained clustering setting. Denoting $\mathcal{S}_n^{\mathrm{F}}$ as the seed superpixels of video V_n, we want to build for $\mathcal{S}_n^{\mathrm{F}}$ a constraint matrix $\mathbf{Z}_n = \{Z_{ij}\}_{N_n \times N_n}$, $N_n = |\mathcal{S}_n^{\mathrm{F}}|$:

$$Z_{ij} = \begin{cases} +1, & (s_i, s_j) \in \mathcal{M} \\ -1, & (s_i, s_j) \in \mathcal{C} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

where $\mathcal{M}$ denotes the set of to-link constraints, and $\mathcal{C}$ denotes the set of not-to-link constraints. The not-to-link constraints forbid two objects that are physically separated to be linked together and are computed based on the following simple spatial relationship:$(s_i, s_j) \in \mathcal{C}$ if there is no path (i.e. a sequence of nodes connected by edges) from s_i to s_j in an adjacency graph built for the seed superpixels of the current frame. To compute $\mathcal{M}$, we select a pair of superpixels (s_i, s_j) that are adjacent in a frame, and warp them to the next and previous 5 frames using the forward and the backward optical flow respectively. If the warped superpixels still remain close to each other, i.e., exhibiting common-fate, (s_i, s_j) are selected to be in $\mathcal{M}$; otherwise, no constraints are assigned to (s_i, s_j).

Since we rely only on the gestalt law of common fate to generate constraints, the graphs are robust to intra-object motion difference arising from self-occlusion or articulated motions. For instance, the different parts of the ostrich are linked together due to the fact they stay connected despite the articulated motions. Evidently, there would still be incorrect constraints, such as those to-link constraints that arise when there are different objects interacting with each other in a single video (e.g. the panda and the toy horse). This is where one needs to use multiple videos to tease out the stable aspect of the foreground appearance.

In order to estimate the background model, we also need to extract the seed superpixels that can represent the background. We used the simple boundary prior proposed by [27], namely, we select those superpixels that reside along the image boundary but do not belong to $\mathcal{S}^{\mathrm{F}}$, and denote the set as $\mathcal{S}^{\mathrm{B}}$.

2.2 Iterative Constrained Clustering and Model Selection

Given N input videos $\mathcal{V} = \{V_1, V_2, ..., V_N\}$, seed superpixels $\mathcal{S}^{\mathrm{F}} = \{\mathcal{S}_1^{\mathrm{F}}, \mathcal{S}_2^{\mathrm{F}}, \ldots, \mathcal{S}_N^{\mathrm{F}}\}$, constraint matrices $\mathcal{Z} = \{\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_N\}$ and an affinity matrix $\mathbf{W} \in \mathcal{R}^{M \times M}$ $(M = \sum_{n=1}^{N} |\mathcal{S}_n^{\mathrm{F}}|)$, which describes the similarity between all seed superpixels, the objective in this subsection is to estimate the number of cluster K and divide the seed superpixels into K clusters, each of which models a foreground object.

To accomplish the objective, we rely on the to-link constraints to provide the necessary prior to bind the non-uniformly colored zones of say, a panda or a leopard together in one cluster. However, recall that some constraints in our $\mathbf{Z}_n$ may be incorrect due to the interaction between different objects or to the errors from the initial foreground-background segmentation. Thus, we need to prune these incorrect constraints to avoid incorrect binding or incorrect separation when clustering the superpixels.

Based on the assumption made in the last subsection, namely, those correct constraints must be stable and recur for all input videos while those incorrect ones should not recur in most videos, we propose an iterative constrained clustering algorithm to deal with the aforementioned issues. Our key idea is similar to the cross-validation procedure, where the incorrect constraints from one matrix $\mathbf{Z}_n$ are detected by finding the inconsistency between the clustering results and the remaining $(N-1)$ constraint matrices. The proposed algorithm is summarized in Algorithm 1.

For the sake of clarity, let's first assume we have two simple input videos. The first video contains a flapping flag with red and white stripes, and the second contains a flapping flag with red, white and blue stripes (in this case, by the end of the process, our extracted foreground must contain only white and red stripes, since they consistently appear in the two videos). In our algorithm, we extract superpixels from the areas of the flags, and create a similarity matrix $\mathbf{W}$ based on color appearance of these superpixels. We generate constraint matrices: $\mathbf{Z}_1$ and $\mathbf{Z}_2$, where $\mathbf{Z}_1$ will provide the links between the red and white superpixels, and $\mathbf{Z}_2$ will provide the links for the red, white and blue superpixels. Our idea here is that

Algorithm 1 Iterative Constrained Clustering and Model Selection

```
Input: $\mathbf{W}$, $\mathcal{Z} = \{\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_N\}$, $T$, CM
  $\mathbf{G}_0 = \mathrm{diag}(\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_N)$;
  for $t = 1 \to T$ do
    $\mathbf{G}_t = \mathbf{G}_{t-1}$;
    for $n = 1 \to N$ do
      Compute the matrix $\mathbf{G}_t^n$ by (3);
      Get the clustering result $L_n$ by (4);
      Compute $R(\mathbf{Z}_n)$, the violated constraint ratio of $\mathbf{Z}_n$;
    end for
    if no violated constraints detected then
      $\mathbf{G}^* = \mathbf{G}_t$ and break;
    end if
    Select the constraint matrix $\mathbf{Z}_n$ with the largest $R(\mathbf{Z}_n)$;
    Update $\mathbf{Z}_n$ by (5) and update $\mathbf{G}_t$ accordingly;
  end for
  Get the model number $K$ and the final clustering result $L^*$ by clustering $\mathbf{W}$ s.t. $\mathbf{G}^*$;
  if CM == 0 then
    Remove the clusters that are not common;
    Adjust $K$ and $L^*$ accordingly;
  end if
  return $K$ and $L^*$
```

if we remove the links in $\mathbf{Z}_1$, and do the clustering on $\mathbf{W}$ subject to $\mathbf{Z}_2$, we end up with the blue stripes included in the foreground. However, if we remove the links in $\mathbf{Z}_2$, and do the clustering on $\mathbf{W}$ subject to $\mathbf{Z}_1$, the extracted foreground will comprise only the white and red stripes grouped together, because there is no link to the blue superpixels. For the latter case, there will be inconsistency between the clustering result and the constraints described by $\mathbf{Z}_2$, necessitating $\mathbf{Z}_2$ to be corrected or updated. Through this verification process, we can prune incorrect constraints and this forms the core of our algorithm.

For more detailed discussion of Algorithm 1, we start with combining all constraints $\mathcal{Z}$ into a matrix $\mathbf{G}_0$, such that $\mathbf{G}_0 = \mathrm{diag}(\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_N)$. In every iteration t, our goal is to select one $\mathbf{Z}_n$ that currently has the highest violated constraint ratio, and then to update it. To achieve this, in the second loop (the n-loop), we first remove each set of constraints one by one. We denote $\mathbf{G}_t^n$ the $\mathbf{G}_t$ with $\mathbf{Z}_n$ removed from the matrix:

$$\mathbf{G}_t^n(i,j) = \begin{cases} 0, & \text{if } (i,j) \in \Omega_n, \\ \mathbf{G}_t(i,j), & \text{otherwise.} \end{cases} \tag{3}$$

where Ω_n is a set of indices of $\mathbf{Z}_n$ that we want to remove.

Next, we perform the following constrained clustering:

$$\text{Cluster on } \mathbf{W}, \text{ s.t. } \mathbf{G}_t^n. \tag{4}$$

In each iteration n, we are interested only in the clustering results of the foreground superpixels, denoted as L_n. Since $\mathbf{Z}_n$ is excluded in this round, it has no effect on L_n. Subsequently, we can compare L_n to $\mathbf{Z}_n$ and record the violated constraints of $\mathbf{Z}_n$. If $\mathbf{Z}_n(i,j) = +1$ but $L_n(i) \neq L_n(j)$, or $\mathbf{Z}_n(i,j) = -1$ but $L_n(i) = L_n(j)$, then we increase the number of violation. The ratio of violated constraint of $\mathbf{Z}_n$, denoted as $R(\mathbf{Z}_n)$, is then computed as the number of violation over the number of all constraints.

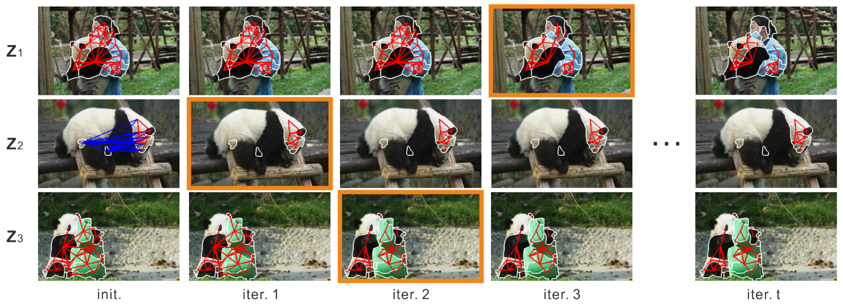


Fig. 4. The constraint updating process by Algorithm 1 on the panda sequences. The constraint graphs having the highest violated constraint ratio and thus selected for update in each iteration are bordered in orange.

Having processed the n-loop, we choose the constraint matrix that has the highest violated constraint ratio, and then update it as follows:

$$\mathbf{Z}_n(i,j) = \begin{cases} 0, & \text{if } \mathbf{Z}_n(i,j) \text{ violates } L_n, \\ \mathbf{Z}_n(i,j), & \text{otherwise.} \end{cases} \tag{5}$$

This will create new configurations of links that are more consistent, since those that are inconsistent with the others are pruned.

Fig. 4 visualizes the constraint updating process by Algorithm 1. It can be seen that the incorrect constraints such as those to-link ones between the panda and the toy horse or those not-to-link ones within the panda are successfully removed iteratively, while the correct and common to-link constraints that connect the panda's white and black patches remain alive.

We stop the iteration when no more violation of constraint is found or the preset maximum iteration limit is reached. The cluster number K and the final clustering result L^* are obtained by running the constrained clustering algorithm based on the final overall constraint matrix $\mathbf{G}^*$. A final minor point is that, as discussed in Sect. 1, one can choose to discard or retain the extraneous objects extracted, depending on the needs. We use the parameter CM for this purpose. In our implementation, we set CM $==0$ which means that the clusters that do not appear in all input videos should be removed. For some applications, if we want to allow some foreground objects to irregularly occur, then the flag CM is set to 1.

Implementation Details We extract the normalized color histogram as the feature descriptors of superpixels as in [23, 16] and compute the pair-wise affinity using the following formula:

$$\mathbf{W}(i,j) = \exp\left\{ -\frac{\|\chi^2(c_i, c_j)\|^2}{\sigma_c} \right\}, \tag{6}$$

where c_i denotes the color histogram and $\chi^2(\cdot,\cdot)$ represents the χ^2-distance between two histograms. To perform constrained clustering, we adopt the Exhaustive and Efficient Constraint Propagation method (EECP) from [28] to incorporate the constraints into the affinity matrix. As EECP does not have an in-house step to perform model selection, we adopt the state-of-the-art SCAMS method from [29] to perform simultaneous clustering and model selection on the modified affinity matrix. Please refer to [28] and [29] for the details of the EECP and the SCAMS algorithms respectively.

2.3 MRF Based Object Segmentation

Assuming the seed superpixels have been clustered by Algorithm 1 into K groups for the foreground, $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_K\}$, we now augment it with the background seed superpixels $\mathcal{S}^{\mathrm{B}}$. We then learn a $K+1$ class SVM classifier that can infer an appropriate distance metric to distinguish the $K+1$ classes. This is done in an one-vs-all scheme by using one of $\mathcal{F}_k$ or $\mathcal{S}^{\mathrm{B}}$ as positive data and the others as negative data. Normalized color histograms are used as the feature descriptors in this step.

Having obtained the appropriate distance metrics for the foreground object models and the background model, we can use them to refine the segmentation results via a graph-cut method [30, 31]. We define a graph over each video's superpixels with nodes representing superpixels and edges between two nodes corresponding to the cost of a cut between two superpixels. Then, we seek to minimize the following energy function for multi-class video segmentation:

$$E(\mathbf{f}) = \sum_{i \in \mathcal{S}} D_i(f_i) + \lambda \sum_{i,j \in \mathcal{N}} V_{i,j}(f_i, f_j) \tag{7}$$

where $\mathbf{f}$ is the label vector of the superpixel nodes with each element $f_i \in [1, K+1]$, and $\mathcal{N}$ defines the spatiotemporal neighborhood of the superpixels.

The data term $D_i(f_i)$ penalizes the labeling of the superpixel x_i with f_i, which is described as $D_i(f_i) = 1 - P_{f_i}(x_i)$, where $P_{f_i}(x_i)$ is the estimated probability of assigning x_i with label f_i, calculated using the learnt one-vs-all SVM for f_i. The smoothness term $V_{i,j}(f_i, f_j)$ encourages the labeling to be spatiotemporally consistent, and is defined as:

$$V_{i,j}(f_i, f_j) = \begin{cases} e^{-(\omega_1 d_c + \bar{\omega}_1 d_f)}, & \text{if } f_i \neq f_j \text{ and } A^s_{ij} = 1, \\ e^{-(\omega_2 d_c + \bar{\omega}_2 d_o)}, & \text{if } f_i \neq f_j \text{ and } A^t_{ij} = 1, \\ 0, & \text{if } f_i = f_j. \end{cases} \tag{8}$$

where $A^s_{ij} = 1$ and $A^t_{ij} = 1$ indicate spatial adjacency and temporal adjacency respectively. The spatial adjacency is only based on the spatial relationship in a single frame, as we want to keep the MRF to a simple pairwise clique. To define temporal adjacency, we warp the superpixels forward and backward to the adjacent frames using optical flow, and then, those superpixels in the adjacent frames that overlap the warped area are selected as the temporal neighbors. The weights w_i and $\bar{w}_i$ ($w_i + \bar{w}_i = 1$) are used to trade off the influence of the color distance and the motion distance. We define the color distance $d_c(i, j)$ as the χ^2-distance between the color histograms of the superpixels, and the motion distance $d_f(i, j)$ between spatially adjacent superpixels as the Euclidean distance

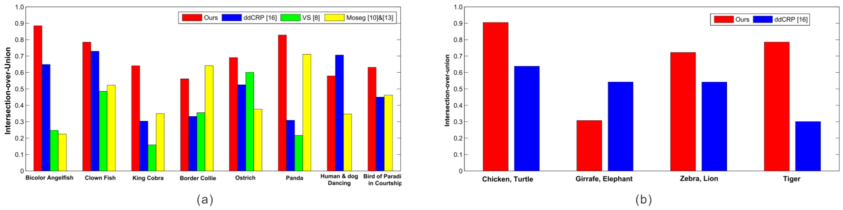


Fig. 5. Comparison of segmentation accuracies on (a) CFViCS and (b) MOViCS.

between the mean motions of the pixels in the superpixels. For temporally adjacent superpixels, their motion distance $d_o(i,j)$ is computed as the average area of two way after-motion overlap, which indicates how likely it is for x_i to move to x_j and vice versa.

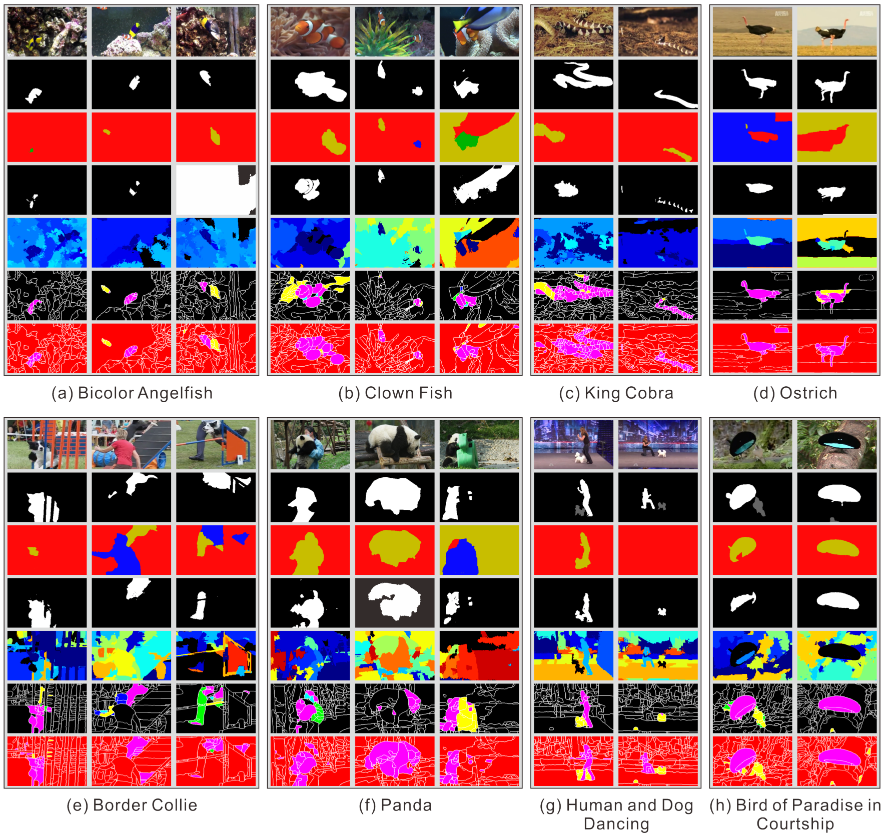
3 Experiments

We applied our method on two datasets and compared the results with those from the state-of-the-art video segmentation methods and video co-segmentation methods. To quantify the results, we employed the intersection-over-union (IOU) metric which is defined as $M(S,G) = \frac{S \cap G}{S \cup G}$, where S is the segmentation result and G is the ground truth. As the video segmentation methods do not link figure objects across videos, we computed their IOU metrics independently in each video and obtain the average as the final IOU figure, while for the video co-segmentation method, the IOU metric was computed jointly in all videos, with S restricted to the segments having the same label. For those videos whose foregrounds have multiple objects, we did not include for comparison those video segmentation methods that can only generate two-layer segmentation.

Exp. on the CFViCS Dataset: We built the Complex Foreground Video Co-Segmentation (CFViCS) dataset that comprises of 8 sets of videos selected to cover the challenges mentioned in Sect. 1. The ground truth of this dataset was manually annotated, and is depicted in the second rows of (a) through (h) of Fig. 6. The CM parameter in Algorithm 1 is set to 0 for this dataset.

Fig. 5(a) depicts the IOU metrics on the CFViCS. It shows that our method achieved the best performance on most of the sequences, and in average outperformed the ddCRP [16] by 20%, the VS [8] by 39%, and the Moseg ([10] postprocessed by [13]) by 25%.

The segmentation results on the CFViCS are shown in Fig. 6. The VS [8] performed poorly in nearly all the sequences in this dataset, and the chief reason was the difficulties in obtaining enough good object proposals when the foreground or background is complex. The video segmentation algorithm obtained by postprocessing [10] with [13] tended to have a good performance when the foreground undergoes rigid motions, as can be seen in the Clown Fish sequences (Fig. 6(b)) and the Border Collie sequences (Fig. 6(e)). However, its performance degraded severely when confronted with articulated motion and inaccurate optical flow estimates (Fig. 6(c) and (d)). In comparison, our method was able to resolve the ambiguity arising from articulated motion and rectify the errors caused by inaccurate optical flow.



(a) Bicolor Angelfish (b) Clown Fish (c) King Cobra (d) Ostrich

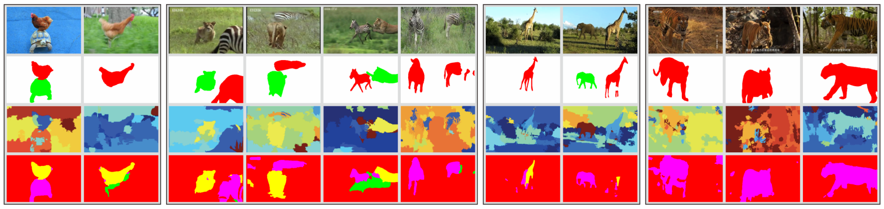
(e) Border Collie (f) Panda (g) Human and Dog Dancing (h) Bird of Paradise in Courtship

Fig. 6. Segmentation results on the CFViCS dataset. In each example, from top to bottom: original video frames, ground truth, results of [10] post-processed by[13], results of [8], results of [16], results of our method after Algorithm 1, and our final results with MRF refinement. Best viewed in color.

The ddCRP method [16] does not organize the segmentation into foreground and background, which to some degree increases the difficulty in matching the foreground across videos. Even if it succeeds in matching, it is likely to over-segment those complex foreground objects with variegated appearance (Fig. 6(e) and (f)). In comparison, it can be seen from the sixth row of Fig. 6 that our iterative constrained clustering and model selection algorithm manages to group different parts of the heterogeneous foreground together.

Exp. on the MOViCS Dataset: We also tested our method on the Multi-Object Video Co-Segmentation (MOViCS) dataset from [16]. This dataset allows the foreground to comprise of objects irregularly occurring in the videos. Thus, for experimental comparison, we set the CM parameter in Algorithm 1 to 1.

The comparison between our method and the ddCRP of [16] on the MOViCS is shown in Fig. 5(b) and Fig. 7. Our method outperformed the ddCRP [16] in three out of four sets of videos, and by 17% in average. Again, as can be seen in Fig. 7, the ddCRP [16] was likely to generate a severely over-segmented results



(a) Chicken, Turtle (b) Zebra, Lion (c) Giraffe, Elephant (d) Tiger

Fig. 7. Segmentation results on the MOViCS dataset. In each example, from top to bottom: original video frames, ground truth, results of [16], and results of our method.

Video set	#GT	#MS	Video set	#GT	#MS
CFViCS					
Bicolor Angelfish	1	2	Human and Dog Dancing	2	2
Border Collie	1	3	Ostrich	1	1
Clown Fish	1	2	Panda	1	1
King Cobra	1	1	Bird of Paradise in Courtship	2	2
MOViCS					
Chicken, Turtle	2	4	Giraffe, Elephant	2	2
Zebra, Lion	2	3	Tiger	1	2

Table 2. The true numbers of objects in the foreground (#GT) and the model selection results of our method (#MS).

on both the foreground and background, whereas our method was able to capture the foreground objects as unified entities.

The model selection results of our method on the CFViCS and the MOViCS are also shown in Table 2. It can be seen that our method obtained correct model selection results in half of the video sets. Even for those incorrect cases, the errors in the model selection were mainly due to some background patches being incorrectly extracted as foreground (see Fig. 6 and Fig. 7). Most of these patches were separated from the true foreground objects. Thus they could be easily removed by some user interaction if necessary.

4 Conclusions

We have presented a video co-segmentation framework for the separation of complex foreground and background. We first perform an initial figure/ground separation using motion cues to obtain seed superpixels and their pairwise constraints. An iterative constrained clustering algorithm is then put forth for model selection and estimation. Finally, a multiclass labeling MRF is used to obtain refined segmentation results. We have tested our method on the CFViCS and the MOViCS datasets; the experimental results demonstrate its success in addressing the challenges present in realistic foreground extraction.

Acknowledgement. This work was partially supported by the Singapore PS-F grant 1321202075 and the grant from the National University of Singapore (Suzhou) Research Institute (R-2012-N-002).

References

1. Spelke, E.S.: Principles of object perception. Cognitive Science **14** (1990) 29–56
2. Wang, M., Ni, B., Hua, X.S., Chua, T.S.: Assistive tagging: a survey of multimedia tagging with human-computer joint exploration. ACM Computing Surveys **44** (2012)
3. Fowlkes, C., Martin, D., Malik, J.: On measuring the ecological validity of local figure/ground cues. In: ECVP. (2003)
4. Maire, M.: Simultaneous segmentation and figure/ground organization using angular embedding. In: ECCV. (2010)
5. Ren, X., Fowlkes, C., Malik, J.: Figure/ground assignment in natural images. In: ECCV. (2006)
6. Stahl, J., Wang, S.: Convex grouping combining boundary and region information. In: ICCV. (2005)
7. Lee, Y., Kim, J., Grauman, K.: Key-segments for video object segmentation. In: ICCV. (2011)
8. Zhang, D., Javed, O., Shah, M.: Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In: CVPR. (2013)
9. Sun, D., Wulff, J., Sudderth, E.B., Pfister, H., Black, M.J.: A fully-connected layered model of foreground and background flow. In: CVPR. (2013)
10. Brox, T., Malik, J.: Object segmentation by long term analysis of point trajectories. In: ECCV. (2010)
11. M.A., P.: Low-level and high-level contributions to figure-ground organization. Oxford University Press (2014)
12. Peterson, M., Gibson, B.: Must figure-ground organization precede object recognition? an assumption in peril. Psychological Science **5** (1994) 253–259
13. Ochs, P., Brox, T.: Object segmentation in video: A hierarchical variational approach for turning point trajectories into dense regions. In: ICCV. (2011)
14. Alexe, B., Deselaers, T., Ferrari, V.: What is an object? In: CVPR. (2010)
15. Endres, I., Hoiem, D.: Category independent object proposals. In: ECCV. (2010)
16. Chiu, W.C., Fritz, M.: Multi-class video co-segmentation with a generative multi-video model. In: CVPR. (2013)
17. Criminisi, A., Cross, G., Blake, A., Kolmogorov, V.: Bilayer segmentation of live video. In: CVPR. (2006)
18. Rahtu, E., Kannala, J., Salo, M., Heikkiä, J.: Segmenting salient objects from images and videos. In: ECCV. (2010)
19. Rother, C., Kolmogorov, V., Minka, T., Blake, A.: Cosegmentation of image pairs by histogram matching incorporating a global constraint into mrfs. In: CVPR. (2006)
20. Vicente, S., Rother, C., Kolmogorov, V.: Object cosegmentation. In: CVPR. (2011)
21. Fu, H., Xu, D., Zhang, B., Lin, S.: Object-based multiple foreground video co-segmentation. In: CVPR. (2014)
22. Wang, C., Guo, Y., Zhu, J., Wang, L., Wang, W.: Video object co-segmentation via subspace clustering and quadratic pseudo-boolean optimization in an mrf framework. IEEE Trans. Multimedia **23** (2014)
23. Joulin, A., Bach, F., Ponce, J.: Multi-class cosegmentation. In: CVPR. (2012)
24. Kim, G., Xing, E.P.: On multiple foreground cosegmentation. In: CVPR. (2012)
25. Galasso, F., Cipolla, R., Schiele, B.: Video segmentation with superpixels. In: ACCV. (2012)

26. Papazoglou, A., Ferrari, V.: Fast object segmentation in unconstrained video. In: ICCV. (2013)
27. Lempitsky, V., Kohli, P., Rother, C., Sharp, T.: Image segmentation with a bounding box prior. In: ICCV. (2009)
28. Lu, Z., Ip, H.: Constrained spectral clustering via exhaustive and efficient constraint propagation. In: ECCV. (2010)
29. Li, Z., Cheong, L.F., Zhou, S.Z.: SCAMS: Simultaneous clustering and model selection. In: CVPR. (2014)
30. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. TPAMI **23** (2001) 1222–1239
31. Cheng, H.T., Ahuja, N.: Exploiting nonlocal spatiotemporal structure for video segmentation. In: CVPR. (2012)