CVPR 2012

Maximum Weight Cliques with Mutex Constraints for Video Object Segmentation

Tianyang Ma
Temple University
tuc09847@temple.edu

Longin Jan Latecki
Temple University
latecki@temple.edu

Abstract

In this paper, we address the problem of video object segmentation, which is to automatically identify the primary object and segment the object out in every frame. We propose a novel formulation of selecting object region candidates simultaneously in all frames as finding a maximum weight clique in a weighted region graph. The selected regions are expected to have high objectness score (unary potential) as well as share similar appearance (binary potential). Since both unary and binary potentials are unreliable, we introduce two types of mutex (mutual exclusion) constraints on regions in the same clique: intra-frame and inter-frame constraints. Both types of constraints are expressed in a single quadratic form. We propose a novel algorithm to compute the maximal weight cliques that satisfy the constraints. We apply our method to challenging benchmark videos and obtain very competitive results that outperform state-of-the-art methods.

1. Introduction and Related Work

Given an unannotated video, our task is to automatically identify the primary object, and segment that object out in every frame. Unsupervised video object segmentation is important for many potential applications, such as activity recognition and video retrieval. Existing methods explore tracking of regions or keypoints over time [4, 6, 22] or perform low-level grouping of pixels from all frames using appearance and motions cues [12, 10]. However, as pointed out in [15], these methods lack an explicit notion of *what a foreground object should look like* in video,and therefore, an ”over-segmentation” result is usually obtained.

Recently, exploring object-centered segmentation in static image has become a very attractive topic, where significant progress has been achieved [9, 7, 1]. In those methods, multiple object hypotheses in form of binary figure-ground segmentation are generated. And the ranking of hypotheses based on their scores implies how plausible these hypotheses are. Using several image cues such as color, texture, and boundary, the model is learned for a generic foreground object, which is then object category independent. An example of object hypothesis produced by the approach [9] is shown in Fig. 2.

By utilizing those figure-ground segmentations with objectness measure, Vicente et al. [27] obtain ”object co-segmentation” from several static images. In contrast to image co-segmentation methods like [24, 26, 14], their method focus more on segmenting *objects* (such as bird or a car).

Lee et al. extend similar idea to video object segmentation in [15]. Instead of only using static objectness measure from [9], dynamic cue is also used to measure how likely a region contains a moving object. They point out that an object region in video should move differently from its surroundings. Specially, their measure compares the optical flow histogram of the region to its surroundings. This does not require any assumptions about camera motion, while being sensitive to different magnitudes of motion. Given the scored regions, top K highest-scoring regions in each frame are collected together to form a region candidate pool $\mathcal{C}$. While many regions in $\mathcal{C}$ belonging to the foreground object, $\mathcal{C}$ may also contain other regions. Similarity based on un-normalized color histogram is computed for every pair of the regions in the pool. Finally, spectral clustering is performed to obtain multiple binary inliers and outliers partitions of the pool. Each cluster (inlier) corresponds to a hypothesis of foreground object regions. Then the obtained clusters are ranked according to the average objectness score of its member regions. The larger is the average score, the more likely a cluster is to contain the primary object in video.

We observe that the region candidate pool $\mathcal{C}$ combines regions across all frames together and discards valuable information of *which* frame each region originates from and *where* it is located in this frame. The proposed approach aims to leverage these information to obtain a better region selection result. We do this by utilizing binary appearance relation between regions in different frames and by enforcing mutual exclusion (mutex) constraints on selected regions. For fair comparison, we adopt the same definition of region objectness in video as [15].

We have the following three insights about selecting primary object regions in video, which make our approach very different from [15]: (1) The selected regions in a cluster should have high objectness score (unary potential) as well as share similar appearance (binary potential) across video frames. This implies the optimal way to select regions is to maximize binary and unary potentials simultaneously, as apposed to [15] in which only binary potentials are considered during spectral clustering. (2) The location of the object in two neighboring frames should be relatively close, considering that the movement of the object is usually smooth. This information is extremely important considering there may be overlap between foreground and background color, which makes the similarity between regions very noisy. (3) We also utilize the common assumption in video segmentation that the primary object appears in every frame. It may change its appearance and shape, due to partial occlusion or self-occlusion, but it is present in each video frame. Therefore, we select exactly one region in every frame as the object region. This prevents the region cluster to be dominated by regions from the same frame, which is very likely to happen, since overlapping regions in a single frame are much more likely to have similar appearance than true object regions in two different frames. Hence, this constraint guarantees that the region selection will not bias to regions in one frame, and pushes the region selection process to discover the true object regions even under significant variations of shape and illumination across the entire video.

We observe that insights (2) and (3) can be expressed as mutex constraints on the object region selection process. They strictly prohibit some regions to appear in the same clique. In particular, insight (3) prohibits two regions from the same frame from belonging to the same clique, and insight (2) prevents two regions from adjacent frames that are relatively far away from belonging to the same clique. We observe that these two constraints cannot be enforced in spectral clustering [21] used in [15].

For our approach to be successful, it is of primary importance that these constraints are strictly enforced. To ensure that this is the case, we propose a novel optimization method. Two example results of our system are shown in Fig. 1. We express the region selection problem as the problem of finding constrained Maximum Weight Cliques (MWCs) in a weighted graph G, where each region corresponds to a node. The diagonal entries of the affinity matrix A of G hold the objectness score of each node. The off diagonal entries represent the appearance similarity between two regions. The maximum weight clique in graph G is the clique with the largest sum of its weights, which means unary potentials and binary potentials are both considered. In our framework, we also constrain the maximum weight clique to satisfy the nonlinear, mutex constraints.

In 1965, Motzkin and Straus [19] established a connection between maximal cliques and the local maximizers of a certain standard quadratic function. Since then, many methods compute MWCs as solutions of the quadratic function relaxed to a simplex. In particular, the approach in [20] has been proven to be a powerful model for many vision problems, such as common pattern discovery [18] and finding stereo correspondence [11]. These approaches compute cliques with the maximum average weight. However, they cannot guarantee that mutex constraints are satisfied.

Recently, [16] introduced a new optimization method that can be interpreted as finding MWCs that satisfy linear equality constraints. This algorithm has built in preference for discrete solutions, and most of the time it converges at a discrete solution which is locally optimal. However, constraints (2) cannot be expressed in a linear equality form, which means that the algorithm in [16] cannot be applied to solve our problem.

To the best of our knowledge, this is the first time video object segmentation is formulated as finding constrained MWCs. Single static image segmentation has been formulated as finding maximal cliques in a very recent paper [13], where maximal cliques are used to compose multiple figure-ground hypotheses into larger interpretations (tilings) of the entire image. Their algorithm adopts a two step solution: step 1 is sequential greedy heuristic, and step 2 is local search heuristic. A similar approach is proposed in [5], where image segmentation is formulated as finding maximum independent set, which is a dual problem to finding maximal cliques. However, in this paper only unary potentials (node weights) are considered.

In Sections 2, 3, and 4, we introduce the edge weights in the region graph, the mutex constraints on regions, and express region selection as finding constrained MWCs, respectively. In Section 5, we utilize the regions selected in Section 4 to achieve a more accurate pixel-level foreground object segmentation. A description of our algorithm for solving constrained MWCs is presented in Section 6, followed by the experimental results in Section 7.

2. Region Graph Construction

Our goal is to segment a foreground object in video without any model of the target. Since we assume no prior knowledge on the size, location, shape or appearance of the target object, we first produce a bag of object ”proposals” in each frame using [9]. The model used in [9] is learned for a generic object from Berkeley Segmentation data, and therefore, it is category independent. Each proposal is a region in the image, an example is shown in Fig 2.

For each frame in the video, we retrieve K regions. (We set $K = 300$ in all experiments.) Given a video consist-

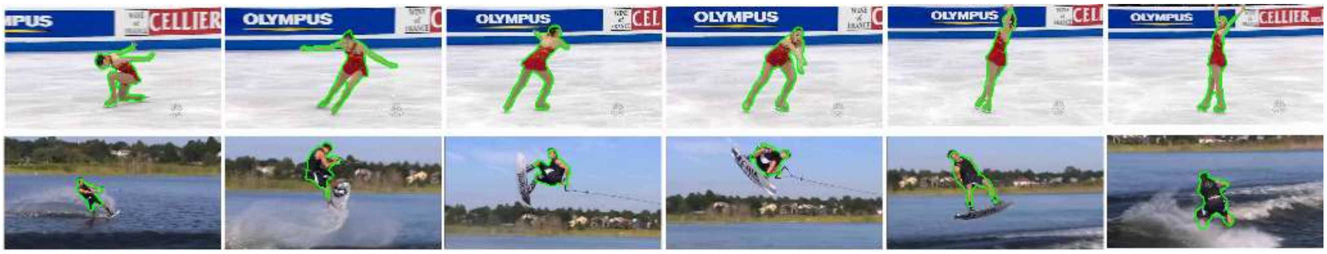

Figure 1. Our object segmentation results on two videos *Yu-Na Kim* and *Waterski* from [10].

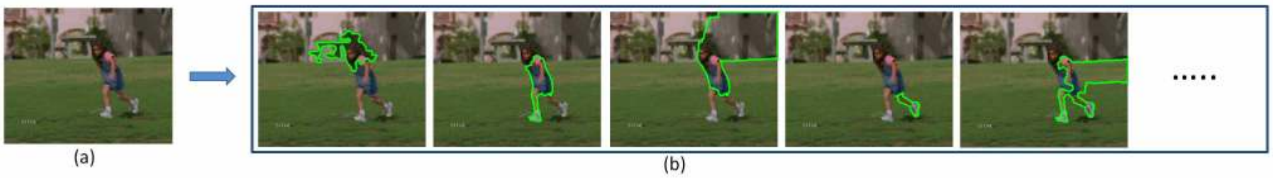

Figure 2. Object proposals produced by [9]. (a) A video frame (b) Proposals ranked in order of ”objectness”.

ing of N frames, we have $K \times N$ regions in total. Our goal is to discover a small subset of regions that contain the same foreground object across all the frames. We construct a weighted graph $G = (V, A)$, in which each node corresponds to one of the $K \times N$ regions, and A is its adjacency matrix. The weight $A(u, u)$ of the node u represents the ”objectness” of the region u, while the weight $A(u, v)$ between two nodes u and v represents the similarity between the two regions. Both are defined below.

We follow the computation of the region ”objectness” in [15]. Specifically, for a region u

$$A(u, u) = ob(u) = sob(u) + mob(u), \tag{1}$$

combines its static intra-frame objectness score $sob(u)$ and motion inter-frame objectness score $mob(u)$. The static score $sob(u)$ is computed using [9]. It reflects the confidence that a region contains a generic object. Several static cues are used to compute this score, such as the probability of a surrounding occlusion boundary, and color differences with nearby pixels.

In [15], the motion objectness $mob(u)$ is introduced to complement to the static score in the case of videos. It measures the confidence that region u corresponds to a coherently moving object in the video. Optical flow histograms are computed for the region u and the pixels $\overline{u}$ around it within a loosely fit bounding box. The score is computed as:

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

where $\chi^2_{flow}(u, \overline{u})$ is the χ^2-distance between L_1-normalized optical flow histograms. The motion score essentially describes how the motion of the region differs from its closest surrounding regions. Both static score and motion score are normalized using the distributions of scores across all regions in the video.

Each region is also described using its Lab color histogram. The similarity between two regions u and v is computed as:

$$A(u, v) = \exp(-\frac{1}{\Omega}\chi^2_{color}(u, v)), \tag{3}$$

where$\chi^2_{color}(u, v)$ is the χ^2-distance between unnormalized color histograms of u and v, and Ω denotes the mean of the χ^2-distance among all the regions. Consequently, if two regions have similar color and similar size, their affinity is high.

3. Mutex Constraints between Regions

One of the key contributions of the proposed work to video segmentation lies in the utilization of hard, mutex (short for mutual exclusion) constraints. They specify which regions cannot be simultaneously selected as part of the segmentation solution. They allow us to eliminate unreasonable configurations of regions, which otherwise have large joint potentials, since both the unary $A(u, u)$ and binary potentials $A(u, v)$ are unreliable. Furthermore, the utilized inference framework allows us to enforce that the solutions satisfy all the constraints. The proposed mutex constraints are based on the following two insights.

Intra-frame mutex constraint: We assume that a true object should appear in *every* frame, and within each frame, only *one* proposal region should be selected. However, the object may be partially occluded or self occluded. This constraint implies that only one object regions candidate produced by [9] is selected for each frame. The same constraint is also utilized in the problem of object co-segmentation from static images [27]. The fact that exactly one object region candidate is selected in each frame is essential for a good selection of candidates mainly for two reasons: 1)

Since many regions in the same frame overlap, their affinities are usually much higher than affinities of true object regions in different frames due to inter-frame variations, such as illumination change. Hence, by excluding affinities of regions from the same frame from consideration in a single clique, the comparison of affinities from different frames becomes more informative. 2) Since we guarantee to select one region for *every* frame, the region selected can be further used as location prior.

Inter-frame proximity constraint: two regions selected in two neighboring frames should be not spatially far away from each other, since the change of the location of the same object in adjacent frames should be smooth.

We encode these two constraints through a binary mutex matrix M defined over all vertices of graph G as

$$M(u,v) = \begin{cases} 1, & \text{if } u \text{ and } v \text{ are in the same frame} \\ & \text{or (if } u \text{ and } v \text{ are in adjacent frames} \\ & \text{and } d(C(u),C(v)) > \tau \text{)} \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

where $C(u)$ and $C(v)$ are the centroid of two regions, and d is their Euclidean distance in pixels. τ reflects the maximum spatial displacement allowed between u and v. We set $\tau = 100$ for all the experiments in order to allow for fast moving objects.

4. Finding Objects as Constrained MWCs

We formulate a region selection problem as finding constrained maximum weight cliques in graph. The input is a weighted graph $G = (V, A)$, where $V = \{v_1, \dots, v_n\}$ is the set of nodes representing the regions in all video frames, n is the number of nodes, and A is a symmetric $n \times n$ affinity matrix with all nonnegative entries, i.e., $A_{ij} \geq 0$ for all $i, j = 1, \dots, n$.

The selected regions are identified with an indicator vector $\mathbf{x} = (x_1, \dots, x_n) \in \{0,1\}^n$, where a given region v_i is selected if and only if $x_i = 1$.

We are also given a symmetric relation $M \subseteq V \times V$ between vertices of the graph. We call M a *mutex* (short for mutual exclusion) relation and represent as binary matrix $M \in \{0,1\}^{n \times n}$. If $M(i,j) = 1$ then the two vertices i, j cannot belong to the same maximum clique. $M(i,i) = 0$ for all vertices i. In other words, mutex represents incompatible vertices that cannot be selected together. Formally, this requirement can be expressed as a constraint on the indicator vector $\mathbf{x} \in \{0,1\}^n$: if $M(i,j) = 1$, then $x_i + x_j \leq 1$. This formulation is equivalent to the requirement $\mathbf{x}^{\mathbf{T}} M \mathbf{x} = 0$.

We obtain the regions of primary object in a given video by solving the following maximization problem

$$\begin{aligned} &\text{maximize } \ f(\mathbf{x}) = \mathbf{x}^T A \mathbf{x} \\ &\text{s.t. } \ \mathbf{x} \in \{0,1\}^n \ \text{ and } \ \mathbf{x}^{\mathbf{T}} M \mathbf{x} = 0. \end{aligned} \tag{5}$$

The goal of (5) is to select a subset of vertices of graph G such that f is maximized and the mutex constraints are satisfied. Since f is the sum of unary and binary affinities of the elements of the selected subset, the larger is the subset, the larger is the value of f. However, the size of the subset is limited by mutex constraints. The problem (5) is a combinatorial optimization problem and is NP-hard [2].

By setting $W = A - \gamma M$ with a sufficiently large γ, we reformulate problem (5) into the following dual form:

$$\begin{aligned} &\text{maximize } \ \mathbf{x}^{\mathbf{T}} W \mathbf{x} = \mathbf{x}^{\mathbf{T}} A \mathbf{x} - \gamma \mathbf{x}^{\mathbf{T}} M \mathbf{x} \\ &\text{s.t. } \ \mathbf{x} \in \{0,1\}^n. \end{aligned} \tag{6}$$

Finally, we relax (6) to

$$\begin{aligned} &\text{maximize } \ \mathbf{x}^{\mathbf{T}} W \mathbf{x} = \mathbf{x}^{\mathbf{T}} A \mathbf{x} - \gamma \mathbf{x}^{\mathbf{T}} M \mathbf{x} \\ &\text{s.t. } \ \mathbf{x} \in [0,1]^n. \end{aligned} \tag{7}$$

In Section 6 an algorithm to solve problem (7) is described. In all video segmentation experiments, we obtained discrete solutions that satisfy all mutex constraints.

Since the maximal clique seeking algorithm we use converges to a local optimum, multiple initializations are required to promise a better performance. We rank the regions in graph G according to their unary score $A(u,u)$, and find the top-L best regions. Each time, we use one region u selected from those top-L best regions to initialize the maximal clique seeking algorithm. We denote the initialization as $\mathbf{x}_{(0)}$, then we set $(\mathbf{x}_{(0)})_u = 1$ and $(\mathbf{x}_{(0)})_i = 0$ for all $i \neq u$. Starting from the $\mathbf{x}_{(0)}$, we obtain a maximal clique indicated by a binary vector $\mathbf{x}^*$. $\mathbf{x}^*$ is a local maximizer of $\mathbf{x}^T A \mathbf{x}$ while satisfying $\mathbf{x}^{*T} M \mathbf{x}^* = 0$.

Therefore, we obtain L maximal cliques in total. We select the best one according to $\mathbf{x}^T A \mathbf{x}$. We find the selected regions as one entries in the indicator vector of this solution. Since the solution satisfies the constraints M defined in Sec 3, we select only one region in each frame, and guarantee every two regions selected in neighboring frames are relatively close to each other. These regions reflect the rough appearance and location of the object in each frame.

5. Foreground Object Segmentation

The obtained segmentation of the object in video in form of selected regions is not very precise. In particular, the segmentation error is lower-bounded by the object region candidates produced by [9]. The error may come from the inaccuracy of the original superpixel extraction or merging. Therefore, we follow the strategy of utilizing the selected regions to learn the appearance model for both foreground and background, e.g., [15, 27]. In addition, we also utilize the location priors. It is particularly easy in our framework, since we have exactly one object region in each frame. Finally, we use GrabCut [23] to infer a more accurate pixel-level object segmentation. For efficiency, rather than labeling pixels in three consecutive frames at once by constructing a space-time graph as in [15], we simply run the GrabCut [23] for each frame separately. This is possible in our framework, since the data term, defined below, which is obtained by our constrained MWCs is very informative.

We denote the pixels in each frame as $S = \{p_1, \dots, p_n\}$, and their labels $f = \{f_1, \dots, f_n\}$, $f_i \in \{0, 1\}$ with 0 for background and 1 for foreground. Then the energy function for minimization is:

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

where $\mathcal{N}$ consists of 8 spatially neighboring pixels.

For the smoothness term V, we use the standard contrast-dependent function defined in [23], which favors assigning the same label to neighboring pixels that have similar color.

Similar to [15], our data term $D_i(f_i)$ defines the cost of labeling pixel p_i with label f_i as:

$$D_i(f_i) = -\log(1 - P_i^c(f_i) \cdot P_i^l(f_i)) \tag{9}$$

where $P_i^c(f_i)$ is the probability of labeling pixel p_i with label f_i based on the appearance (color) cues, $P_i^l(f_i)$ is the probability based on location prior. Both are defined below.

To compute $P_i^c(f_i)$, we first estimate two Gaussian Mixture Models (GMM) in RGB color space to model the foreground (fg) and background (bg) appearance. Since the color may vary significantly over the video frames, we need to learn the color models over all video frames, which is an easy task since we have the object regions inferred as the constrained MWCs. The foreground GMM model fg^{color} is learned from pixels in the regions selected in the constrained MWCs computation. The background GMM model bg^{color} is learned from pixels contained in the complement of selected regions in all the frames. Then given these two color distributions fg^{color} and bg^{color}, we define for each pixel p_i:

$$P_i^c(f_i) = \begin{cases} P(p_i | fg^{color}), & \text{if } f_i = 1 \\ P(p_i | bg^{color}), & \text{if } f_i = 0 \end{cases} \tag{10}$$

For the computation of location probability $P_i^l(f_i)$, we utilize the object regions selected in the constrained MWCs. Given the selected region (we have only one region per frame), we first compute its distance transform. Let $d(p_i)$ denotes the distance of pixel p_i to the selected object region. We compute

$$P_i^l(f_i) = \begin{cases} \exp(-\frac{d(p_i)}{\sigma}), & \text{if } f_i = 1 \\ 1 - \exp(-\frac{d(p_i)}{\sigma}), & \text{if } f_i = 0 \end{cases} \tag{11}$$

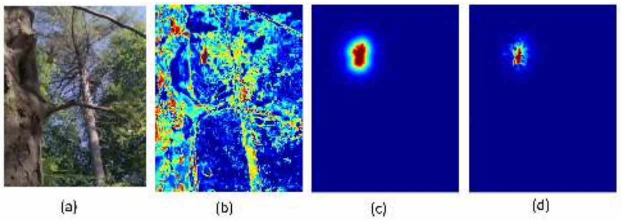

Figure 3. (a) A single frame and the probabilities of the foreground object $f_i = 1$. (b) Color prob. $P_i^c(f_i)$. (c) Location prob. $P_i^l(f_i)$. (d) The joint foreground prob. $P_i^c(f_i) \cdot P_i^l(f_i)$

where σ indicates the confidence of the location prior, the larger is σ, the lower is the confidence. We compute $P_i^c(f_i) \cdot P_i^l(f_i)$ as the probability of foreground (f_i = 1) and background (f_i = 0). As illustrated in Fig 3(b), the color probability is not particularly informative in a global scale of the whole frame, and the main information comes from the possibilty map of the location shown in Fig. 3(c). However, the color information is informative if constrained by the location probability as illustrated by the joint probability shown in Fig 3(d).

After obtaining the data term D and smoothness term V, we use the popular method in [3] to find the optimal f that minimizes the energy function (8), and obtain the final foreground objects in each video frame.

6. Algorithm Description

In this section, we introduce a novel algorithm for finding the constrained MWCs. $f(\mathbf{x}) = \mathbf{x}^{\mathbf{T}} W \mathbf{x}$ denotes the objective function of Eq. (7).

Our algorithm visits a sequence of continuous points $\{\mathbf{y}_{(k)} \in [0,1]^n\}_{k=1,2,\dots}$. In each iteration k, we have two steps. First, given $\mathbf{y}_{(k)}$, for any point $\mathbf{y} \in [0,1]^n$ in its neighborhood, we compute the first-order Taylor approximation of $f(\mathbf{y})$ as

$$\begin{aligned} f(\mathbf{y}) &\approx f(\mathbf{y}_{(k)}) + 2(\mathbf{y} - \mathbf{y}_{(k)})^T W \mathbf{y}_{(k)} \\ &= 2\mathbf{y} W \mathbf{y}_{(k)} - f(\mathbf{y}_{(k)}) \end{aligned} \tag{12}$$

Since the second term $f(\mathbf{y}_{(k)})$ in (12) does not depend on $\mathbf{y}$, the first-order Taylor approximation of $f(\mathbf{y})$ only depends on $\mathbf{y} W \mathbf{y}_{(k)}$, which is a linear function of $\mathbf{y}$. This fact allows an easy computation of a discrete maximizer

$$\tilde{\mathbf{x}}_{(\mathbf{k})} = \arg\max_{\mathbf{y} \in [0,1]^n} \mathbf{y}^T W \mathbf{y}_{(k)} \tag{13}$$

as

$$(\tilde{\mathbf{x}}_{(\mathbf{k})})_i = \begin{cases} 1, & \text{if } (W \mathbf{y}_{(k)})_i > 0 \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

In the second step of iteration k, we want to verify whether the obtained $\tilde{\mathbf{x}}_{(\mathbf{k})}$ can be accepted as a valid discrete solution that increases f. In the case that $f(\tilde{\mathbf{x}}_{(\mathbf{k})}) >$

$f(\mathbf{y}_{(k)})$, we let $\mathbf{y}_{(k+1)} = \tilde{\mathbf{x}}_{(\mathbf{k})}$. In the case that $f(\tilde{\mathbf{x}}_{(\mathbf{k})}) \le f(\mathbf{y}_{(k)})$, we estimate the local maximizer of f in the continuous domain by performing a line search, i.e., by maximizing one dimensional function $h(\alpha) = f(\mathbf{y}_{(k)} + \alpha(\tilde{\mathbf{x}}_{(\mathbf{k})} - \mathbf{y}_{(k)}))$ over the line segment from $\tilde{\mathbf{x}}_{(\mathbf{k})}$ to $\mathbf{y}_{(k)}$. It is easy to show that $h(\alpha)$ obtains its maximum at α defined in (15). It can also be shown that $0 < \alpha < 1$, which guarantees that line search will not reach outside the cube.

$$\alpha = -\frac{(\tilde{\mathbf{x}}_{(\mathbf{k})} - \mathbf{y}_{(k)})^T W \mathbf{y}_{(k)}}{(\tilde{\mathbf{x}}_{(\mathbf{k})} - \mathbf{y}_{(k)})^T W (\tilde{\mathbf{x}}_{(\mathbf{k})} - \mathbf{y}_{(k)})} \tag{15}$$

Then we set $\mathbf{y}_{(k+1)} = \mathbf{y}_{(k)} + \alpha(\tilde{\mathbf{x}}_{(\mathbf{k})} - \mathbf{y}_{(k)})$

Our algorithm stops when the following *stop condition* holds for all coordinates i of vector $\mathbf{x}^* = \mathbf{y}_{(k+1)}$:

$$\begin{aligned} &\text{if } \ (W\mathbf{x}^*)_i > 0, \quad \text{then } \ \mathbf{x}_i^* = 1 \\ &\text{if } \ (W\mathbf{x}^*)_i < 0, \quad \text{then } \ \mathbf{x}_i^* = 0 \end{aligned} \tag{16}$$

We observe that $W\mathbf{x}^* = \frac{1}{2}\nabla f(\mathbf{x}^*)$. Hence $(W\mathbf{x}^*)_i > 0$ means that the direction of the increase of f coincides the direction of ith coordinate, while $(W\mathbf{x}^*)_i < 0$ means that the direction of the increase of f is opposite to the direction of ith coordinate. Therefore, the stop condition tells us that $f(\mathbf{x}^*)$ already has the maximum possible value for every increase direction of f. In other words, we cannot increase f without leaving our domain $[0,1]^n$, meaning that $\mathbf{x}^*$ is a local maximum of f over $[0,1]^n$.

We assume that the initial assignment $\mathbf{y}_{(0)}$ satisfies the mutex constraints, i.e., $\mathbf{y}(0)^T M\mathbf{y}(0) = 0$. This implies that $f(\mathbf{y}_{(0)}) \ge 0$, since all entries in A are non-negative.

The proposed algorithm is summarized in the following pseudo code:

Algorithm 1

```
Input: Matrix $W$, $f(\mathbf{y}_{(0)}) \ge 0$, and $\epsilon > 0$
repeat
    Use (14) to find $\tilde{\mathbf{x}}_{(\mathbf{k})} = \arg\max_{\mathbf{y}\in[\mathbf{0},\mathbf{1}]^{\mathbf{n}}} \mathbf{y}\mathbf{W}\mathbf{y}_{(k)}$
    if $\tilde{\mathbf{x}}_{(\mathbf{k})} = \mathbf{y}_{(k)}$ then
        $\mathbf{y}_{(k+1)} = \tilde{\mathbf{x}}_{(\mathbf{k})}$
    else if $f(\tilde{\mathbf{x}}_{(\mathbf{k})}) > f(\mathbf{y}_{(k)})$ then
        $\mathbf{y}_{(k+1)} = \tilde{\mathbf{x}}_{(\mathbf{k})}$
    else
        Use (15) to compute $\alpha$.
        $\mathbf{y}_{(k+1)} = \mathbf{y}_{(k)} + \alpha(\tilde{\mathbf{x}}_{(\mathbf{k})} - \mathbf{y}_{(k)})$
    end if
until $\mathbf{y}_{(k+1)}$ satisfies (16) or $f(\mathbf{y}_{(k+1)}) - f(\mathbf{y}_{(k)}) < \epsilon$
Output: $\mathbf{y}_{(k+1)}$
```

In all experimental results in the next section, all solutions are discrete. Thus, the proposed algorithm does not require any postprocessing to obtain discrete solutions. We have also verified experimentally that all obtained solutions satisfy the mutex constraints.

7. Experimental Results

We first examine our method on the SegTrack dataset [25]. There are six videos (*monkeydog, bird, girl, birdfall, parachutte, penguin*). For each video, a pixel-level segmentation ground-truth is provided for the primary foreground object. This enables a statistical evaluation of our method. Object segmentation in these videos are extremely challenging due to several facts, such as the primary object are with large shape deformation and foreground and background color has overlap. Same as [15], we do not evaluate our method on *penguin* video since only a single penguin is labeled as the foreground object among a group of penguins.

Given a video, we first produce [9] 300 object candidate regions per frame. We use Lab space histograms to describe color for each region. Each Lab channel has 20 bins. For the color model of the foreground and background, we use RGB color space, and two GMMs with 5 component are learned. Same as [15], we describe motion using optical flow histograms computed from [17] with 60 bins per x and y direction. The region's bounding box is dilated by 30 pixels when computing the background histograms. To initialize the maximal clique computation, each time we select one from the best 50 object regions candidates according to $A(u,u) = ob(u)$. We set $\sigma = 20$ for the computation of $P_i^l(f_i)$. In the graph cut energy function (8), $\delta = 1$ in all our experiments.

Due to the efficiency of the proposed constrained MWCs algorithm, on a PC with 3.4Ghz and 8GB RAM, it only takes 2 minutes to select regions by constrained MWCs with 50 different initializations. The binary graph cut on single frame takes about 0.1s in average.

We compare the results with three state-of-the-art methods [15], [25] and [8]. The method in [15] and our method are unsupervised. They automatically discover the primary object in image as well as segment the object out. The methods in [25] and [8] require minor supervision with the object labeled in the first frame. The results are shown in Table 1. Our method has the lowest average per frame segmentation error over the 5 test videos. It also achieves the lowest segmentation error on 3 out of 5 videos. Compared to [15], which also does not require manual object initialization, we achieve better results on 4 out of 5 videos. Some segmentation results are shown in Fig. 4.

The results in Table 1 report the average per-frame, pixel error rate computed in comparison to the ground-truth segmentation. Specially, it is computed as [25]:

$$error = \frac{\mathbf{XOR}(f, GT)}{F} \tag{17}$$

where f is the label for every pixel in a given video, GT is the ground-truth label, and F is the total number of frames in a given video. Since all videos are roughly of the same size, the average error rate over the 5 videos is computed

Video (No. frames)	Ours	[15]	[25]	[8]
birdfall (30)	**189**	288	252	454
cheetah (29)	**806**	905	1142	1217
girl (21)	1698	1785	**1304**	1755
monkeydog (71)	**472**	521	563	683
parachute (51)	221	**201**	235	502
Average	**542**	592	594	791
Manual seg.:	No	No	Yes	Yes

Table 1. Segmentation error as measured by the average number of incorrect pixels per frame. Lower values are better.

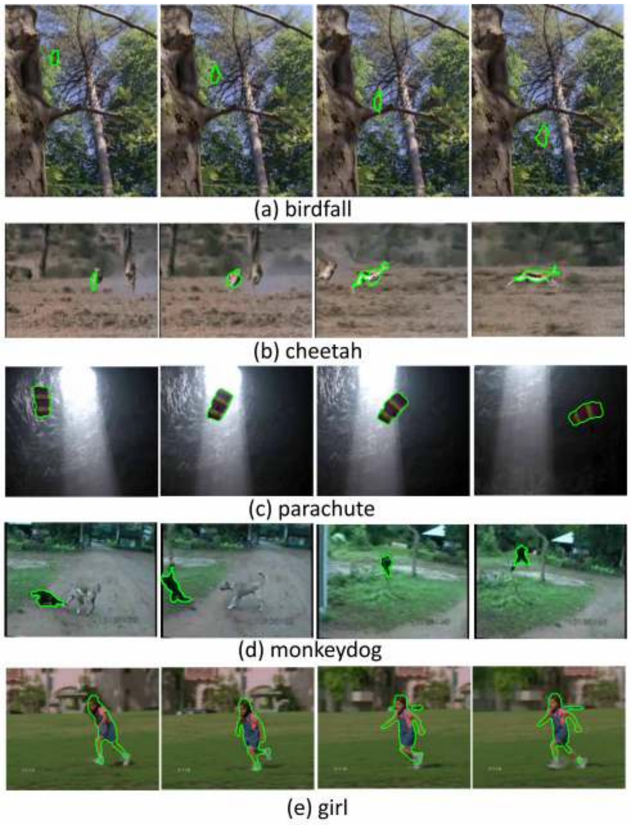

Figure 4. Segmentation results. Best viewed in color.

as average over all frames in all videos, i.e., we treat all 5 videos as a single video and apply (17).

As we mentioned above, even without the pixel-based object segmentation described in Secion 5, the object regions selected by constrained MWCs in Section 4 alone can be regarded as the segmentation result. In Table 2, we report the pixel error of the constrained MWCs regions segmentation results, although it is lower-bounded by the accuracy of the region candidates produced by [9]. The lower-bound error is computed as the error of the region candidate with the lowest error as compared to the ground-truth pixels. This reflects the lowest segmentation pixel error we could achieve by only selecting regions from computing the constrained MWCs.

	Ours	constrained MWC	Lower bound
birdfall	189	311	295
cheetah	806	1258	700
girl	1698	3063	2973
monkeydog	472	497	493
parachute	221	803	680

Table 2. Segmentation error comparison. We compare our entire proposed method (Ours) to the region segmentation results obtained by the region selection as constrained MWCs. The lower bound error is the lowest possible error of regions produced by [9].

	constrained MWC	w/o constraints
birdfall	311	589
cheetah	1258	1772
girl	3063	3742
monkeydog	497	2024
parachute	803	883

Table 3. Segmentation error comparison of the constrained MWCs optimization with and without the mutex constraints.

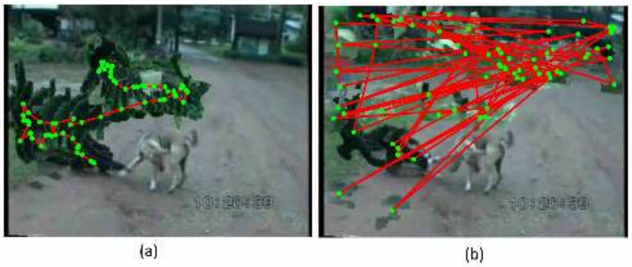

Figure 5. The trajectories of centroids of selected regions, green dots connected with red lines, overlaid over the first frame: (a) when inter-frame mutex constraints are used and (b) when inter-frame mutex constraints are *not* used.

We can see that, for videos *birdfall, monkeydog*, the results are very good merely using regions selected by constrained MWCs. Moreover, with the exception of *cheetah*, the pixel error is rather close to the lower bound. This demonstrates that the proposed region selection scheme as constrained MWCs is a powerful tool for video segmentation.

As shown in Table 3, the segmentation error increases significantly if inter-frame proximity mutex constraints, which express spatiotemporal coherency, are not taken as input to the constrained MWC optimization. We also provide a visual illustration of the importance of these mutex constraints in Fig. 5. We compare the trajectories of the constrained MWCs region centroids computed with and without this mutex constraints. They are shown overlaid over the first video frame. We can see that with the constraints, the trajectory of the centroid is very smooth, and the selected regions are always focusing on the primary object, i.e., the monkey in the example video. This shows that the mutex constraints significantly increase the robustness of the constrained MWCs optimization. They allow us to eliminate unreasonable region selection hypotheses, which result from unreliable region affinity relations, and consequently, play a critical role in selecting correct object regions.

We also examine our method on two videos *Yu-Na Kim* and *Waterski* from [10]. While [10] focus on labeling every pixel in image using motion and appearance cues, we automatically identify the primary object, i.e., ice skater and water skier, and segment them out in every frame. Qualitative results are shown in Fig 1.

8. Conclusions

We present a novel method for video object segmentation. It utilizes mutex constraints in order to obtain reliable segmentations of foreground object under large variations of shape, appearance, and illumination. The selection of object regions is performed simultaneously for all frames of the video. The computation is cast as finding maximum weight cliques in the region graph. We propose a novel algorithm for solving this problem. Since it yields discrete solutions in all presented experimental results, it did not require any postprocessing to obtain discrete solutions. We have also verified experimentally that all obtained solutions satisfy the mutex constraints.

9. Acknowledgment

We would like to thank Ian Endres and Derek Hoiem for releasing their software of generating object proposals [9], and to Olga Veksler for providing the optimization toolbox for graph cuts [3]. We would also like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper. This work received support from the NSF under Grants IIS-0812118, BCS-0924164, and OIA-1027897.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *CVPR*, 2010. 1

[2] Y. Asahiro, R. Hassin, and K. Iwama. Complexity of finding dense subgraphs. *Discrete Applied Mathematics*, 121:15 – 26, 2002. 4

[3] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11), 2001. 5, 8

[4] W. Brendel and S. Todorovic. Video object segmentation by tracking regions. In *ICCV*, 2009. 1

[5] W. Brendel and S. Todorovic. Segmentation as maximum-weight independent set. In *NIPS*, 2010. 2

[6] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, 2010. 1

[7] J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *CVPR*, 2010. 1

[8] P. Chockalingam, S. N. Pradeep, and S. Birchfield. Adaptive fragments-based tracking of non-rigid objects using level sets. In *ICCV*, 2009. 6, 7

[9] I. Endres and D. Hoiem. Category independent object proposals. In *ECCV*, 2010. 1, 2, 3, 4, 6, 7, 8

[10] M. Grundmann, V. Kwatra, M. Han, and I. A. Essa. Efficient hierarchical graph-based video segmentation. In *CVPR*, 2010. 1, 3, 8

[11] R. Horaud and T. Skordas. Stereo correspondence through feature grouping and maximal cliques. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(11), 1989. 2

[12] Y. Huang, Q. Liu, and D. N. Metaxas. Video object segmentation by hypergraph cut. In *CVPR*, 2009. 1

[13] A. Ion, J. Carreira, and C. Sminchisescu. Image segmentation by figure-ground composition into maximal cliques. In *ICCV*, 2011. 2

[14] A. Joulin, F. R. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In *CVPR*, 2010. 1

[15] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *ICCV*, 2011. 1, 2, 3, 5, 6, 7

[16] M. Leordeanu, M. Hebert, and R. Sukthankar. An integer projected fixed point method for graph matching and map inference. In *NIPS*, 2009. 2

[17] C. Liu. Beyond pixels: Exploring new representations and applications for motion analysis. *Doctoral Thesis. Massachusetts Institute of Technology.*, 2009. 6

[18] H. Liu and S. Yan. Common visual pattern discovery via spatially coherent correspondences. In *CVPR*, 2010. 2

[19] T. Motzkin and E. Straus. Maxima for graphs and a new proof of a theorem of turan. *Canad. J. Math*, 1965. 2

[20] M. Pavan and M. Pelillo. Dominant sets and pairwise clustering. *PAMI*, 29:167-172, 2007. 2

[21] P. Perona and W. T. Freeman. A factorization approach to grouping. In *ECCV*, 1998. 2

[22] A. V. Reina, S. Avidan, H. Pfister, and E. L. Miller. Multiple hypothesis video segmentation from superpixel flows. In *ECCV*, 2010. 1

[23] C. Rother, V. Kolmogorov, and A. Blake. ”grabcut”: interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3), 2004. 5

[24] C. Rother, T. P. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of image pairs by histogram matching - incorporating a global constraint into MRFs. In *CVPR*, 2006. 1

[25] D. Tsai, M. Flagg, and J. M. Rehg. Motion coherent tracking with multi-label MRF optimization. In *BMVC*, 2010. 6, 7

[26] S. Vicente, V. Kolmogorov, and C. Rother. Cosegmentation revisited: Models and optimization. In *ECCV*, 2010. 1

[27] S. Vicente, C. Rother, and V. Kolmogorov. Object cosegmentation. In *CVPR*, 2011. 1, 3, 5