Key-Segments for Video Object Segmentation

Yong Jae Lee, Jaechul Kim, and Kristen Grauman
University of Texas at Austin
yjlee0222@utexas.edu, jaechul@cs.utexas.edu, grauman@cs.utexas.edu

Abstract

We present an approach to discover and segment foreground object(s) in video. Given an unannotated video sequence, the method first identifies object-like regions in any frame according to both static and dynamic cues. We then compute a series of binary partitions among those candidate "key-segments" to discover hypothesis groups with persistent appearance and motion. Finally, using each ranked hypothesis in turn, we estimate a pixel-level object labeling across all frames, where (a) the foreground likelihood depends on both the hypothesis's appearance as well as a novel localization prior based on partial shape matching, and (b) the background likelihood depends on cues pulled from the key-segments' (possibly diverse) surroundings observed across the sequence. Compared to existing methods, our approach automatically focuses on the persistent foreground regions of interest while resisting over-segmentation. We apply our method to challenging benchmark videos, and show competitive or better results than the state-of-the-art.

1. Introduction

Video object segmentation is the problem of automatically segmenting the objects in an unannotated video. While the unsupervised form of the problem has received relatively little attention, it is important for many potential applications including video summarization, activity recognition, and video retrieval.

Existing unsupervised methods explore tracking regions or keypoints over time [4, 30, 5] or formulate clustering objectives to group pixels from all frames using appearance and motion cues [11, 10]. Aside from the well-known challenges associated with tracking (drift, occlusion, and initialization) and clustering (model selection and computational complexity), these methods lack an explicit notion of *what a foreground object should look like* in video data. Consequently, the low-level grouping of pixels usually results in a so-called "over-segmentation".

Instead, we propose an approach that automatically discovers a set of *key-segments* to explicitly model likely foreground regions for video object segmentation. Our main idea is to leverage both static and dynamic cues to detect persistent object-like regions, and then estimate a complete segmentation of the video using those regions and a novel localization prior that uses their partial shape matches across the sequence. See Figure 1.

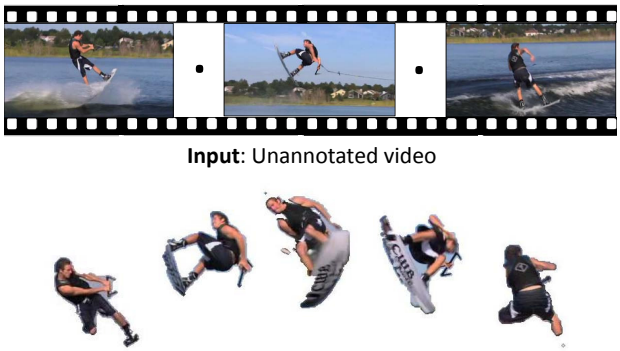

Figure 1. Our idea is to discover a set of key-segments to automatically generate a foreground object segmentation of the video.

To implement this idea, we first introduce a measure that reflects a region's likelihood of belonging to a foreground object. To capture *object-like motion and persistence*, we use dynamic inter-frame properties such as motion difference from surroundings and recurrence. Intuitively, a region that moves differently from its surroundings and appears frequently throughout the video will likely be among the main objects of interest. Conversely, one that seldom occurs is more likely to be an uninteresting, background object. To capture *object-like appearance and shape*, we use static properties such as a well-defined closed boundary in space and clear separation from surroundings, as recently explored in static images [8, 6, 1]. We use both aspects to group the key-segments, estimating multiple inlier/outlier partitions of the candidate regions. Each ranked partition automatically defines a foreground and background model, with which we solve for a pixel-wise segmentation using graph cuts on a space-time MRF. The rank reflects the corresponding object's centrality to the scene.

How does key-segment discovery help video object segmentation? The key-segments are a reliable source for

learning the appearance of a foreground object, since they were determined to be both object-like and frequently occurring in the video. Furthermore, key-segments detected across the sequence imply probability distributions for the location and scale of the object in other frames, which we show how to capture through a novel partial shape matching localization prior. What is the advantage of having a *group* of key-segments? An ensemble alleviates imprecise segmentations on any individual key-segment and captures background diversity in the video, since the background visible in each key-segment's frame can vary. In practical terms, our approach substantially reduces annotator effort; rather than outlining an object of interest, one can simply use (or peruse) the suggested foreground object(s).

Contributions Our main contribution is an automatic approach for segmenting foreground objects discovered in video. To our knowledge, no prior work explores category-independent foreground segmentation for videos where simple background subtraction is insufficient. Towards this goal, important novel components of our technique include (1) a new motion-based measure of object-like regions in video that complements existing image-based cues, (2) a localization prior using partial shape matches in video, and (3) a space-time graph segmentation that accommodates the key-segments. We apply our unsupervised method to challenging benchmark videos, analyze its components in detail, and show state-of-the-art results compared to existing unsupervised and supervised methods.

2. Related Work

We review prior work along two major themes: interesting region detection, and video object segmentation.

Detecting probable foreground regions Finding "interesting" objects in image or video is a long-standing topic in vision, addressed in various forms including saliency detection, figure-ground segmentation, or object discovery. Whereas most saliency detectors rely on bottom-up image cues (e.g., [12, 9]), recent work suggests that higher-level saliency may actually be *learned* from labeled data of segmented objects [19, 1, 6, 8], drawing on classic Gestalt cues. In particular, interesting approaches to generate and rank an image's multiple figure-ground segmentation hypotheses are explored in [6, 8], with results showing that higher-ranked figure proposals are more likely to be objects in an image. Inspired by this premise, we expand the notion of "object-like" regions to video, and introduce the requisite motion and persistence cues.

Beyond single images, some work considers discovering repeated patterns among pairs or *groups* of unlabeled images [26, 13, 16]. It is challenging since some unknown portion of any image may contain the repeated pattern, calling for iterative refinement techniques [13] or graph-based segmentation of discovered objects [26, 16]. Video offers stronger temporal consistency constraints than assorted snapshots, which our approach aims to leverage.

In video with a stationary background, moving foreground regions pop-out well with classic background subtraction algorithms (e.g., [28]). However, for generic videos with unknown camera motion, lighting changes, and poor resolution—or interesting but static objects!—they are inadequate. Repeated features in video are extracted in [18, 23]; however, the local feature approach means the objects are often not delineated well from background, whereas we seek fully segmented regions. More importantly, the grouping objective does not explicitly target discovery of a salient object. To our knowledge, no prior work considers ranking category-independent "object-like" foreground regions in video, as we do in this work.

Video object segmentation Video object segmentation is often performed in an interactive or supervised way. Interactive methods require a user to annotate object boundaries in some key frames, which are then propagated to other frames while a user stands by to adjust errors [2, 22, 32]. Tracking-based methods attempt to reduce the supervision to a manual segmentation on only the first frame (e.g., [24, 29]). However, all such methods demand user input drawing regions of interest, and may suffer from sensitivity to a user's annotation expertise.

Bottom-up approaches can segment videos in a fully automatic manner, based on cues like motion and appearance similarity. Motion segmentation methods (e.g., [27]) cluster pixels in video using bottom-up motion cues. Recent methods either perform pixel-level segmentation in a spatio-temporal video volume from scratch [10], begin with an image segmentation per frame and then match segments across nearby frames, e.g., [11, 4, 30], or use dense flow to cluster long-term motion trajectories [5]. Without any top-down notion of objects, however, such methods tend to over-segment, yielding regions that taken alone may lack semantic meaning.

Shape provides a strong grouping cue for object parts of disparate appearance. Extensive work on weakly-supervised object segmentation integrates top-down shape priors, and some are applied to sequence data [15, 31]. In contrast, we propose to segment generic objects (of unknown categories) in video, with neither direct user interaction nor provided category exemplars. Beyond being unsupervised, our shape-based prior is novel in that local partial shape matches are used to prime object localization in earlier/later frames of the video.

3. Approach

Our goal is to discover object-like *key-segments* in an unlabeled video, and learn appearance and shape models from them to automatically segment the foreground objects.

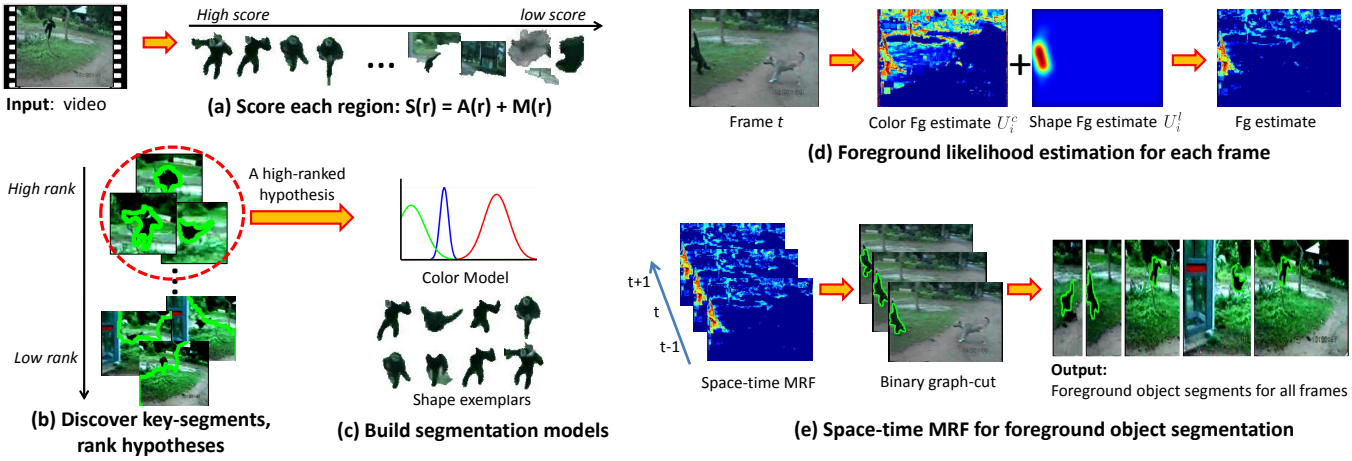

Figure 2. Algorithm overview. See ordered steps (a) through (e).

There are three main steps to our approach: (1) scoring each image region using appearance and motion cues to determine how likely it is to belong to a foreground object; (2) clustering the regions to discover key-segments that represent a single object, and ranking those clusters according to their region scores; and (3) segmenting each foreground object in the video using its model learned from the corresponding key-segments. The final output is a ranked set of foreground object segmentations. We now describe each step in turn.

3.1. Finding Object-like Regions in Video

In order to segment a foreground object in the video, we first need a representation of that object. Since we assume no prior knowledge on its size, location, shape, or appearance, we initially generate a diverse set of object "proposals" in each frame using the static region-ranking method of [8]. The proposals are guided by models learned from true segmentations of arbitrary objects[1], and have been shown to better align with object boundaries than traditional bottom-up segments do. For each frame in the video, we generate roughly 1000 regions.

To find "object-like" regions among the proposals, we look for regions that have (1) appearance cues typical to objects in general, and (2) differences in motion patterns relative to their surroundings. These properties are well-suited for defining objects in video; any region that is salient in terms of both appearance and motion may correspond to a true object. Specifically, we define a function:

$$S(r) = A(r) + M(r), \tag{1}$$

that scores a region r according to its static intra-frame appearance score $A(r)$ and dynamic inter-frame motion score $M(r)$. See Figure 2 (a).

[1]Note those exemplars are *disjoint* from the objects appearing in the videos we process; specifically, the region proposal function of [8] was trained with Berkeley Segmentation data.

We compute $A(r)$ using [8]. It reflects cues indicative of a generic object, such as the probability of a surrounding occlusion boundary, color differences with nearby pixels, and the probability of belonging to a vertical surface. Note this measure only looks at the appearance of the region within each frame, and does not care about the motion.

We compute $M(r)$ to measure the confidence that region r corresponds to a coherently moving object in the video. We compute optical flow histograms for the region r and the pixels $\bar{r}$ around it within a loosely fit bounding box, and then score r as:

$$M(r) = 1 - \exp(-\chi^2_{flow}(r, \bar{r})), \tag{2}$$

where $\chi^2_{flow}(r, \bar{r})$ is the χ^2-distance between L_1-normalized optical flow histograms. Note that this cue is not simply looking for large motions or appearance changes from background (e.g., as one would in background subtraction). Rather, we are describing how the motion of the proposal region differs from its closest surrounding regions; this allows us to forgo assumptions about camera motion, and also to be sensitive to different magnitudes of motion. Furthermore, the region r itself is a product of an object-like ranking, not an arbitrary bottom-up segment.

Before combining $A(r)$ and $M(r)$, we rescale each to standard Gaussians using the distribution of scores across all regions in the video.

3.2. Discovering Key-Segments Across Frames

Given the scored regions, we next identify *groups* of *key-segments* that may represent a foreground object in the video. For each frame, we take the top N highest-scoring regions to form a candidate pool $\mathcal{C}$ spanning the entire sequence. Many regions belonging to a foreground object should be present in $\mathcal{C}$ (as they were predicted to be most "object-like"), but there may also be noisy segments. Thus, we specifically treat this stage as gathering multiple hypotheses among the highly ranked object-like regions, com-

puting multiple partitions of $\mathcal{C}$. In Section 3.3 we explain how to use them to segment the entire video.

To extract the groups, we first define similarity between two regions r_m and r_n:

$$K(r_m, r_n) = \exp(-\frac{1}{\Omega}\chi^2_{color}(r_m, r_n)), \quad (3)$$

where $\chi^2_{color}(r_m, r_n)$ is the χ^2-distance between unnormalized color histograms of r_m and r_n, and Ω denotes the mean of the χ^2-distances among all regions. This measure gives high affinity to regions that have similar color and similar size. We compute the pairwise affinities between all regions m,$n \in \mathcal{C}$, to obtain the affinity matrix $K_\mathcal{C}$.

We next perform a form of spectral clustering [21, 20] with $K_\mathcal{C}$ to produce multiple binary inlier/outlier partitions of the data, with the objective of maximizing the inliers' intra-cluster affinity (normalized by the number of inliers). Each eigenvector of $K_\mathcal{C}$ produces a partitioning of the data; we binarize the continuous eigenvector to form an indicator vector that denotes the inlier set, using the technique in [20].

Each cluster (inlier set) is a hypothesis h of a foreground object's key-segments. We automatically rank the clusters based on the average object-like score $S(r)$ of its member regions. If that scoring is successful, the clusters among the highest ranks will correspond to the primary foreground object(s), since they are likely to contain frequently appearing object-like regions (as we confirm in Figure 5 below). See Figure 2(a-b) for a summary of the pipeline so far.

3.3. Foreground Object Segmentation

Each ranked partition ("key-segment hypothesis") automatically defines a foreground and background model. For now, suppose we extract a color distribution and set of shape exemplars for each hypothesis (see Figure 2(c)). We next devise a space-time Markov Random Field (MRF) model that uses these models to guide a pixel-wise segmentation for the entire video. In practice, we process the hypotheses in rank order, exploiting the quality of the object-like ranking discussed above.

Importantly, a top-ranked hypothesis helps form models of both the object itself *and* the remaining background objects, for two reasons. First, the foreground features common to the selected key-segments are more pronounced, while unique or isolated features are discounted. Second, the diversity in background appearance is captured through the (potentially) different backgrounds present in each key-segment's frame. For example, as the camera pans to follow a primary object of interest, the surrounding background can change substantially; so long as a key-segment hypothesis spans frames from various backgrounds, it will help propagate the figure-ground labeling accordingly.

Space-time graph definition We define a graph over each frame's pixels: a node corresponds to a pixel, and an edge between two nodes corresponds to the cost of a cut between two pixels. The energy function we minimize for hypothesis h takes a familiar form:

$$E(f, h) = \sum_{i \in \mathcal{S}} D^h_i(f_i) + \gamma \sum_{i,j \in \mathcal{N}} V_{i,j}(f_i, f_j), \quad (4)$$

where f is a labeling of the pixel nodes, $\mathcal{S} = \{p_1, \ldots, p_n\}$ is the set of n pixels in the video, $\mathcal{N}$ consists of neighboring pixels, and i and j index the pixels. Each pixel p_i is assigned to $f_i \in \{0, 1\}$, where 0 corresponds to background and 1 corresponds to foreground. The pixel neighborhood $\mathcal{N}$ consists of four spatially neighboring pixels in the same frame, and two temporally neighboring pixels in adjacent frames. We assign a pixel's temporal neighbor in the next frame by its optical flow vector displacement. Related space-time graphs are defined in [29, 30].

The neighborhood term $V_{i,j}$ encourages label smoothness in space and time. We use a standard contrast-dependent function defined in [25], which favors assigning the same label to neighboring pixels that have similar color.

The data term D^h_i defines the cost of labeling pixel i with label f_i, given key-segments in h. Specifically,

$$D^h_i(f_i) = -\log\left(\alpha \cdot U^c_i(f_i, h) + (1 - \alpha) \cdot U^l_i(f_i, h)\right), \quad (5)$$

where $U^c_i(\cdot)$ is the color-induced cost, and $U^l_i(\cdot)$ is the local shape match-induced cost. Both terms are depicted in Figure 2(d), and explained in detail next.

Appearance-based models To model the fg and bg appearance, we estimate two Gaussian Mixture Models (GMM) in RGB colorspace: (1) a GMM fg^{color} for pixels in h's key-segments; and (2) a GMM bg^{color} for pixels in the complement of h's key-segments, among all frames in h. We set $U^c_i(f_i, h)$ to be the pixel-likelihoods computed from each GMM. A pixel that has similar color to the foreground (background) object will have high cost if labeled as background (foreground).[2]

Location priors via partial shape matching Beyond simple appearance terms, for video segmentation, we also want to exploit the consistency of recurring foreground objects viewed over time. In particular, we have a strong localization prior from one frame to the next. Our use of optical flow to define neighbors (see Sec. 3.3) partially captures this via label smoothness, but is closely tied to appearance agreement and can fail when the foreground and background GMMs share similar color components. Thus, as the final component of our model, we introduce a novel technique to prime the location and scale of the foreground object in a frame using key-segment shapes.

The main idea is to use the key-segments detected across the sequence, projecting their shapes into other frames via local shape matching. The spatial extent of that projected

[2] Note the $-\log(\cdot)$ in Eqn. 5.

shape then serves as a location and scale prior in which we prefer to label pixels as foreground. Since we have multiple key-segments and many possible local shape matches, many such projected shapes are aggregated together, essentially "voting" for the location/scale likelihoods. See Figure 3.

More specifically, we project the key-segments onto each frame in the video by matching Boundary Preserving Local Regions (BPLR) [14]. A BPLR is a densely-extracted local feature that preserves object boundaries and partial shape.[3] For each video frame, we generate BPLRs and retain for shape matching those that produce better (lower distance) matches to the BPLRs of the key-segments than to the BPLRs of their image complements. We create a vote space that has the same size as the frame, and project the matched key-segment onto the frame after aligning the locations and scales of the matched BPLRs. We weight the votes according to the match similarity. This process is repeated for all retained BPLRs, and we normalize the vote space such that the maximum value is one.

Then, the vote value at p_i gives its fg location likelihood:

$$U_i^l(f_i) = \begin{cases} P(p_i|bg^{shape}(h)), & \text{if } f_i = 0; \\ P(p_i|fg^{shape}(h)), & \text{if } f_i = 1, \end{cases} \quad (6)$$

and the bg location likelihood is its complement. $U_i^l(f_i)$ measures whether a pixel lies in a projected region of the key-frames. Pixels that are part of a commonly projected region will have high probability of being labeled as foreground[2]. See "Shape Fg estimate" in Figure 3.

When is this most useful? By using partial (local) shape feature matches to drive each shape projection, we intend to account for deformations and articulations that the foreground object may exhibit. For example, a running monkey's global shape can vary significantly from frame-to-frame. However, its arms and legs will only undergo small changes in shape. Thus, a local match (e.g., at the arm or leg) derived from a key-segment can usefully map in the rough global shape prior, despite the change in pose.

In addition, this likelihood helps disambiguate labels when there are similar colors in both the fg and bg models, or if there is a background object that did not appear in any of the key-segments' frames. Note that the key-segment color models only capture cues *within* their own frames. This means that the background objects that appear in the non-key-segment frames are not modeled, and may easily be mislabeled as foreground. For example, if a tree with brown leaves appears behind a brown monkey (the fg object), the tree could otherwise be mislabeled as foreground. Table 2 in the results specifically validates the impact of the term $U_i^l(f_i)$.

Minimization procedure for video labeling We minimize Eqn. 4 with binary graph cuts [3], and use the resulting label assignment as the foreground object segmentation of the video for hypothesis h. See Figure 2(e).

[3]Other descriptors are feasible, but we specifically choose BPLR due to its robustness when matching deformable objects.

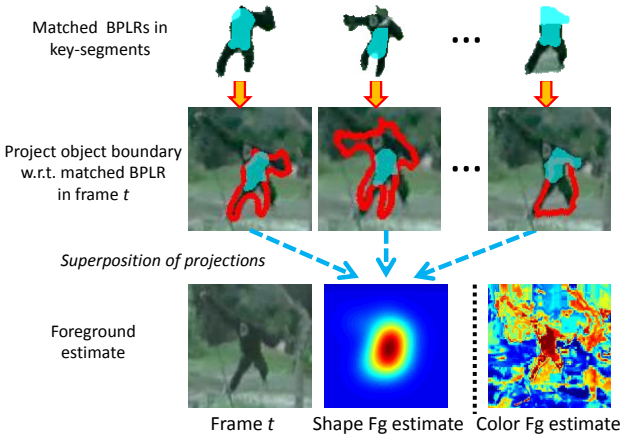

Figure 3. Fg location and scale estimates with BPLR matches.

For efficiency, rather than segment the entire video at once, we sequentially label each frame in turn, using a space-time graph of three frames that connects its two adjacent frames. In addition, for better accuracy, rather than simply pass through the frames in sequential order, we proceed in a greedy ordering from the most confident frames that contain key-segments. That is, we start by labeling and fixing the key-segments' frames, and then solve others in their order of temporal proximity. This more effectively propagates the fg/bg labels of one frame to the next through optical flow connections.

3.4. Summary of the Approach

To recap, our method takes an unlabeled video, and produces foreground-background segmentations ranked by the object's expected centrality to the scene. The main steps are: (1) extract proposal regions from all frames, (2) score all regions by $S(r)$, (3) take top-ranked regions, and partition into inlier/outlier hypotheses. For each hypothesis, (4) extract foreground model and local shape features from all its key-segments, (5) match shape features across all frames to create shape-based foreground likelihood maps, (6) minimize Eqn. 4 using graph cuts with series of space-time graphs, (7) return binary pixel-wise segmentation.

Since our method ranks the foreground results by confidence, one can use it in a completely unsupervised manner to define the primary foreground objects (e.g., for summarization). Alternatively, if a user is in the loop, s/he can select the desired foreground object.

4. Results

The main questions in our experiments are (1) to what extent are object-like regions better identified by using motion cues unique to video, (2) how well does our method rank each hypothesis, and (3) how accurate is our method's

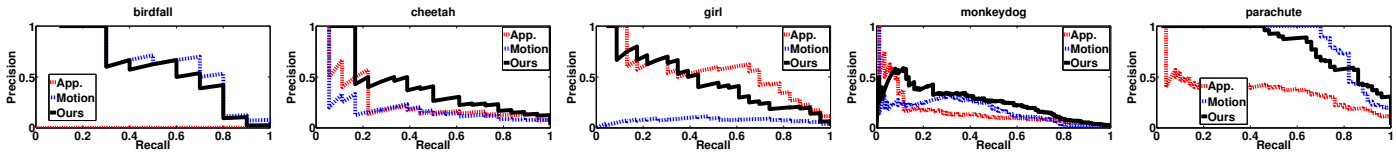

Figure 4. Precision-Recall curves for foreground object prediction. We analyze the different components of our video object-like scoring function. (**Ours**): full model; (**App.**): appearance-based region scoring; (**Motion**): motion-based region scoring. Higher curves are better.

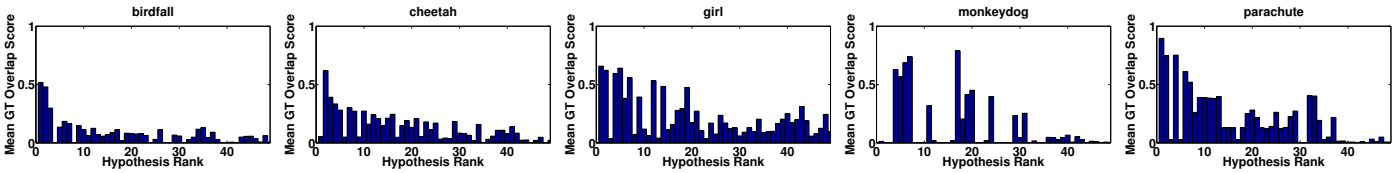

Figure 5. Our ranked hypotheses and their mean ground-truth overlap scores. Our ranking focuses attention to primary foreground objects.

foreground object segmentation?

Dataset: We test on two datasets: [29] and [10], eight videos in total. We use the SegTrack dataset [29], which contains six videos (*monkeydog, bird, girl, birdfall, parachute, penguin*) and pixel-level ground-truth (GT) for the primary foreground object. The videos span a wide degree of difficulty with challenges such as fg/bg color overlap, large shape deformation, and large camera motion. To our knowledge, it is the largest publicly available pixel-labeled video dataset. We do not provide in-depth quantitative results on the *penguin* video, since it lacks the GT to properly evaluate our algorithm; only a single penguin is labeled as the foreground object amidst a group of penguins.

In addition, we generate qualitative results on two videos from the dataset of [10]; note that it lacks pixel-level ground-truth needed for quantitative analysis.

Implementation details: We use [8] to generate regions. To describe color, we use Lab space histograms, with 23 bins per channel, and $K = 5$ component GMMs. To describe motion, we use optical flow histograms with 61 bins per x and y direction, using [17]; we dilate a region's bounding box by 30 pixels when computing the background histograms. We extract BPLRs every 6 pixels. We set $N = 10$.

For the graph-cuts minimization, we set $\alpha = 0.5$ and $\gamma = 4$ for the smoothness term. These parameters are fixed for scoring all videos. We smooth the partial shape match vote space with a Gaussian kernel to be robust to minor alignment errors and shape deformations.

Generating regions takes about 3 minutes / frame, computing GMMs takes about 2 minutes, and segmentation takes about 1 second / frame with a Matlab implementation.

Object prediction accuracy We first evaluate our method's ability to predict object-like regions, and compare: (1) the static appearance component [8] that computes $A(r)$, (2) the dynamic motion component $M(r)$, and (3) our full model $S(r)$ that uses both.

Figure 4 shows precision-recall curves for the three variants on all regions in each video. A region r is considered to be a true positive (i.e., foreground object), if its *overlap score* $= \frac{|GT \cap r|}{|GT \cup r|}$ is greater than 0.5.

The results clearly demonstrate that motion plays a significant role in identifying foreground object regions in video. This is particularly true for the *birdfall* and *parachute* sequences, in which the foreground object has large motion patterns compared to its surroundings. Static appearance is important as well, as can be seen for the *girl, cheetah*, and *monkey* videos. In the *girl* video, the foreground object exhibits articulated motions in which one part (e.g., arm) has substantially larger motion compared to another part (e.g., torso), which explains the low precision of the motion-only component. By accounting for both motion and appearance, our full model produces the best predictions overall.

Object hypothesis rank accuracy We next evaluate our method's hypothesis ranking. Figure 5 shows the mean ground-truth region overlap score for each of the ranked hypotheses. High rank hypotheses have high mean overlap-scores, while low rank hypotheses have low mean overlap-scores. This shows our automatically generated ranking is highly indicative of how well each hypothesis represents the primary object of interest. Among all videos, only the *monkeydog* sequence lacks a strong hypothesis among the top three ranks. This is due to an artifact of the data: each frame contains black margins, which artificially produce high motion scores (since their motion is constant, while the remaining objects are moving or appear to be moving due to camera motion); the top-three hypotheses predict these to be the foreground object. However, the fourth ranked hypothesis correctly predicts the monkey to be the primary object.

What do the hypotheses and their key-segments look like? Figure 6 shows key-segments of the highest-ranked hypothesis that corresponds to the primary object. The number in parentheses indicate its rank. On six of the eight videos, our very top-ranked hypothesis corresponds to the primary foreground object. If desired, one could easily re-rank the hypotheses to enforce diversity by penalizing pixel

	Ours	[29]	[7]	Top $A(r)$ region	Bg Sub
birdfall	288	**252**	454	26156	7435
cheetah	**905**	1142	1217	27728	28763
girl	1785	**1304**	1755	10236	45019
monkeydog	**521**	563	683	38083	31099
parachute	**201**	235	502	75168	27242
penguin	136285(*)	**1705**	6627	147686	61089
Manual seg?	No	Yes	Yes	No	No

Table 1. Segmentation error as measured by the average number of incorrect pixels per frame. Lower values are better. We compare our method (**Ours**) with two state-of-the-art methods ([29] and [7]), which require the first frame to be annotated. $*$*See text about penguin ground-truth.*

overlap with higher ranked key-segments.

It is evident that the key-segments are representative exemplars of the foreground object. This allows our method to learn reliable color and shape models for segmenting out the object in all frames, including those that did not produce any key-segments, as we show next.

Object segmentation accuracy In this section, we evaluate our method's final segmentation results. We compare against two state-of-the-art methods: (1) the motion coherence segmentation method of [29], and (2) the level set-based tracker of [7]. These methods require human labeling of the object boundary in the first frame. In contrast, our method requires no hand drawn supervision to guide the segmentation. (One may choose among our method's ranked segmentation proposals, but this does not change segmentation quality.)

Table 1 shows the results. To quantify segmentation accuracy, we use the average per-frame pixel error rate [29], $\epsilon(S) = \frac{|\mathbf{XOR}(S,GT)|}{F}$, where S is each method's segmentation, GT is the ground-truth segmentation, and F is the total number of frames. We evaluate our method with the segmentation of the hypothesis that corresponds to the object with ground-truth annotation.

Our method produces the best results on three of the five videos (*cheetah, monkeydog, parachute*), and produces the second best result on the *birdfall* video. Our higher error on the *girl* video is caused by an over-segmentation of the key-segments. This is primarily due to some inaccurate initial region proposals from [8], which is reasonable since the object exhibits large appearance variation. For the *penguin* video, our top-ranked hypothesis corresponds to the group of penguins, whereas the ground-truth annotates only a single penguin. Since the group of penguins are so close and similar, it is not clear whether one or all penguins makes a better foreground estimate.

The last two columns in Table 1 show error rates when taking the region with the highest appearance-based score $A(r)$ per frame and when performing standard background subtraction [28], respectively. Clearly, $A(r)$ alone is insufficient to predict the primary object in the video. Background subtraction completely falls apart, since it cannot handle large camera motions. By taking into account both motion and persistence to discover the key-segments, we obtain significantly better foreground segmentations.

	Ours	Ours w/o partial shape match
birdfall	**288**	414
cheetah	**905**	1024
girl	1785	**1534**
monkeydog	**521**	1261
parachute	201	**188**

Table 2. Segmentation error. Lower values are better. We compare our full method (**Ours**) with a baseline that only models color information (**Ours w/o partial shape match**). Our partial shape matching improves segmentation quality.

Figure 6 shows qualitative segmentation examples. Our method produces high quality segmentations of the primary foreground object. There are some failure cases as well, such as when the object is mislabeled due to low contrast with its surrounding regions (see last column of *bird* video), and when parts of the object are missed (see the second and third columns of *girl* video).

The last row shows a comparison to the unsupervised method of [10]. Our method produces a figure-ground segmentation at the object-level by automatically finding its key-segments. In contrast, [10] relies only on bottom-up pixel-level motion and appearance cues, which sometimes results in an over-segmentation of an object.

Impact of partial shape matching Finally, we study the impact of our partial shape matching location prior. We compare against a baseline that only models color, but otherwise follows the same pipeline as our full method. For this baseline, we set $\gamma = 50$ as in [25] to adjust the scales of the cost values between the methods. Table 2 shows the results. The partial shape matching improves segmentation accuracy in most videos. As discussed earlier, some of the key-segments of the *girl* video are over-segmented, which means that the projected shape can miss the articulated body parts (e.g., arms); increasing the color term helps in this case. Overall, we find a substantial advantage from the partial shape match.

Conclusion We developed an algorithm that automatically discovers key-segments and groups them to predict the foreground objects in a video. We introduced a novel partial shape match location prior that primes the foreground object's location and scale in each frame. By discovering object-like key-segments, we overcome the limitations of previous bottom-up unsupervised methods that often over-segment an object, and obtain similar or higher quality segmentation than state-of-the-art supervised methods with minimal human input.

Acknowledgements Many thanks to Ian Endres for sharing his code. This research was sponsored in part by DARPA Mind's Eye W911NF-10-2-0059, LLNL B594497, and CSSG N11AP20004.

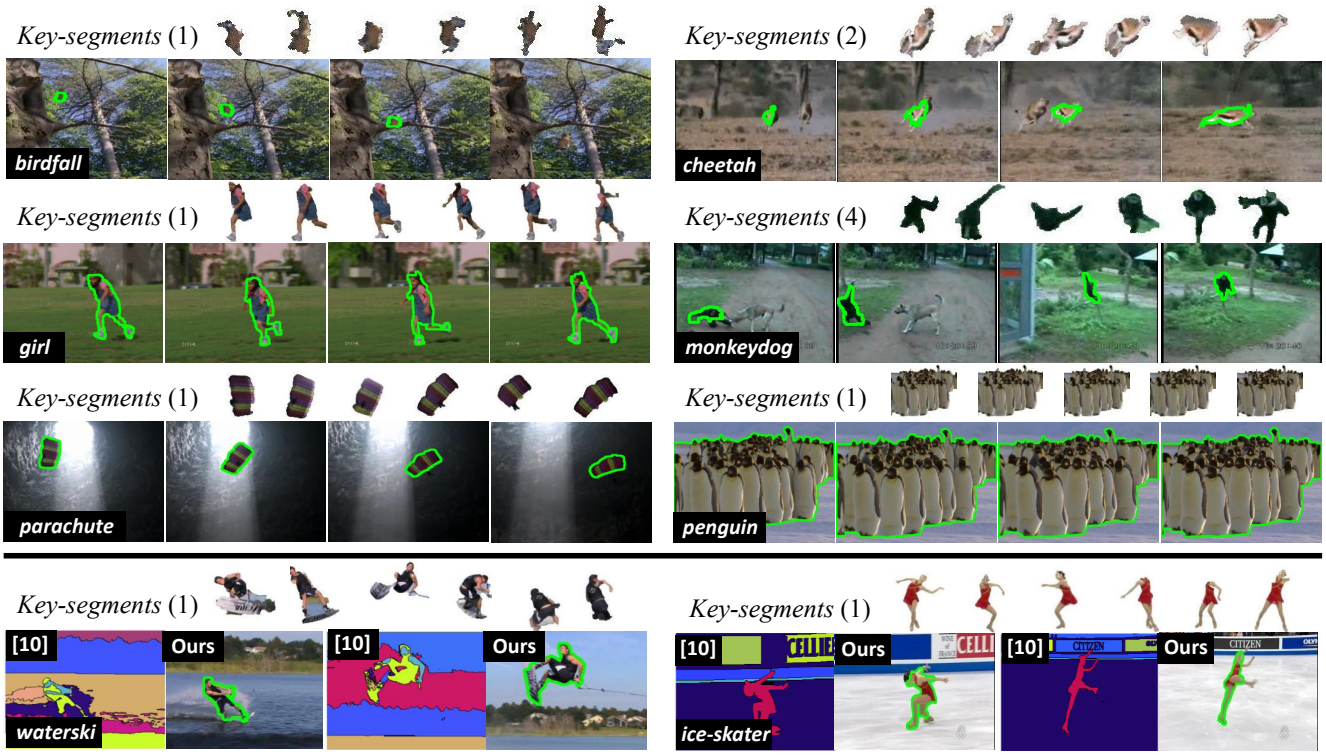

Figure 6. Key-segments and corresponding segmentation results. The numbers indicate the rank of each hypothesis. The hypothesis corresponding to the primary object has high rank, and its key-segments have high overlap with true object boundaries. The first three rows show results on SegTrack [29] videos. The last row compares our results to [10]. Best viewed on pdf.

References

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

[2] X. Bai, J. Wang, D. Simons, and G. Sapiro. Video Snapcut: Robust Video Object Cutout using Localized Classifiers. In *SIGGRAPH*, 2009.

[3] Y. Boykov, O. Veksler, and R. Zabih. Efficient Approximate Energy Minimization via Graph Cuts. *TPAMI*, 20(12):1222–1239, November 2001.

[4] W. Brendel and S. Todorovic. Video Object Segmentation by Tracking Regions. In *ICCV*, 2009.

[5] T. Brox and J. Malik. Object Segmentation by Long Term Analysis of Point Trajectories. In *ECCV*, 2010.

[6] J. Carreira and C. Sminchisescu. Constrained Parametric Min-Cuts for Automatic Object Segmentation. In *CVPR*, 2010.

[7] P. Chockalingam, N. Pradeep, and S. Birchfield. Adaptive Fragments-Based Tracking of Non-Rigid Objects Using Level Sets. In *ICCV*, 2009.

[8] I. Endres and D. Hoiem. Category Independent Object Proposals. In *ECCV*, 2010.

[9] D. Gao, V. Mahadevan, and N. Vasconcelos. The Discriminant Center-Surround Hypothesis for Bottom-Up Saliency. In *NIPS*, 2007.

[10] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient Hierarchical Graph Based Video Segmentation. In *CVPR*, 2010.

[11] Y. Huang, Q. Liu, and D. Metaxas. Video Object Segmentation by Hypergraph Cut. In *CVPR*, 2009.

[12] L. Itti and P. F. Baldi. A Principled Approach to Detecting Surprising Events in Video. In *CVPR*, 2005.

[13] G. Kim and A. Torralba. Unsupervised detection of regions of interest using iterative link analysis. In *NIPS*, 2009.

[14] J. Kim and K. Grauman. Boundary-Preserving Dense Local Regions. In *CVPR*, 2011.

[15] M. P. Kumar, P. H. S. Torr, and A. Zisserman. OBJ CUT. In *CVPR*, 2005.

[16] Y. J. Lee and K. Grauman. Collect-Cut: Segmentation with Top-Down Cues Discovered in Multi-Object Images. In *CVPR*, 2010.

[17] C. Liu. *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis*. PhD thesis, MIT, 2009.

[18] D. Liu and T. Chen. DISCOV: A Framework for Discovering Objects in Video. In *MM*, 2008.

[19] T. Liu, J. Sun, N. Zheng, X. Tang, and H. Shum. Learning to Detect a Salient Object. In *CVPR*, 2007.

[20] E. Olson, M. Walter, J. Leonard, and S. Teller. Single Cluster Graph Partitioning for Robotics Applications. In *RSS*, 2005.

[21] P. Perona and W. Freeman. A Factorization Approach to Grouping. In *ECCV*, 1998.

[22] B. Price, B. Morse, and S. Cohen. Livecut: Learning-based Interactive Video Segmentation by Evaluation of Multiple Propagated Cues. In *ICCV*, 2009.

[23] T. Quack, V. Ferrari, and L. V. Gool. Video Mining with Frequent Itemset Configurations. In *CIVR*, 2006.

[24] X. Ren and J. Malik. Tracking as Repeated Figure/Ground Segmentation. In *CVPR*, 2007.

[25] C. Rother, V. Kolmogorov, and A. Blake. Grabcut. In *ACM Transactions on Graphics*, 2004.

[26] C. Rother, V. Kolmogorov, T. Minka, and A. Blake. Cosegmentation of Image Pairs by Histogram Matching - Incorporating a Global Constraint into MRFs. In *CVPR*, 2006.

[27] J. Shi and J. Malik. Motion Segmentation and Tracking Using Normalized Cuts. In *ICCV*, 1998.

[28] C. Stauffer and W. Grimson. Adaptive background mixture modelsfor real-time tracking. In *CVPR*, 1999.

[29] D. Tsai, M. Flagg, and J. M.Rehg. Motion Coherent Tracking with Multi-Label MRF Optimization. In *BMVC*, 2010.

[30] A. Vazquez-Reina, S. Avidan, H. Pfister, and E. Miller. Multiple Hypothesis Video Segmentation from Superpixel Flows. In *ECCV*, 2010.

[31] Z. Yin and R. Collins. Shape Constrained Figure-Ground Segmentation and Tracking. In *CVPR*, 2009.

[32] J. Yuen, B. Russell, C. Liu, and A. Torralba. Labelme Video: Building a Video Database with Human Annotations. In *ICCV*, 2009.