Video Object Segmentation through Spatially Accurate and Temporally Dense Extraction of Primary Object Regions

Dong Zhang[1], Omar Javed[2], Mubarak Shah[1]
[1]Center for Research in Computer Vision, University of Central Florida, Orlando, FL 32816
[2]SRI International, Princeton, NJ 08540
dzhang@cs.ucf.edu, omar.javed@sri.com, shah@crcv.ucf.edu

Abstract

In this paper, we propose a novel approach to extract primary object segments in videos in the 'object proposal' domain. The extracted primary object regions are then used to build object models for optimized video segmentation. The proposed approach has several contributions: First, a novel layered Directed Acyclic Graph (DAG) based framework is presented for detection and segmentation of the primary object in video. We exploit the fact that, in general, objects are spatially cohesive and characterized by locally smooth motion trajectories, to extract the primary object from the set of all available proposals based on motion, appearance and predicted-shape similarity across frames. Second, the DAG is initialized with an enhanced object proposal set where motion based proposal predictions (from adjacent frames) are used to expand the set of object proposals for a particular frame. Last, the paper presents a motion scoring function for selection of object proposals that emphasizes high optical flow gradients at proposal boundaries to discriminate between moving objects and the background. The proposed approach is evaluated using several challenging benchmark videos and it outperforms both unsupervised and supervised state-of-the-art methods.

1. Introduction & Related Work

In this paper, our goal is to detect the primary object in videos and to delineate it from the background in all frames. Video object segmentation is a well-researched problem in the computer vision community and is a prerequisite for a variety of high-level vision applications, including content based video retrieval, video summarization, activity understanding and targeted content replacement. Both fully automatic methods and methods requiring manual initialization have been proposed for video object segmentation. In the latter class of approaches, [2, 15, 23] need annotations of object segments in key frames for initialization.

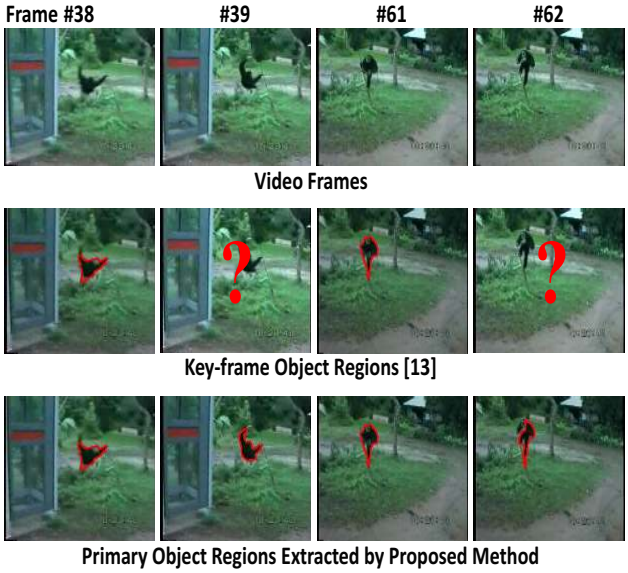

Figure 1. Primary object region selection in the object proposal domain. The first row shows frames from a video. The second row shows key object proposals (in red boundaries) extracted by [13]. "?" indicates that no proposal related to the primary object was found by the method. The third row shows primary object proposals selected by the proposed method. Note that the proposed method was able to find primary object proposals in all frames. The results in row 2 and 3 are prior to per-pixel segmentation. In this paper we demonstrate that temporally dense extraction of primary object proposals results in significant improvement in object segmentation performance. Please see Table 1 for quantitative results and comparisons to state of the art.[Please Print in Color]

Optimization techniques employing motion and appearance constraints are then used to propagate the segments to all frames. Other methods ([16, 20]) only require accurate object region annotation for the first frame, then employ region tracking to segment the rest of frames into object and background regions. Note that, the aforementioned semi-automatic techniques generally give good segmenta-

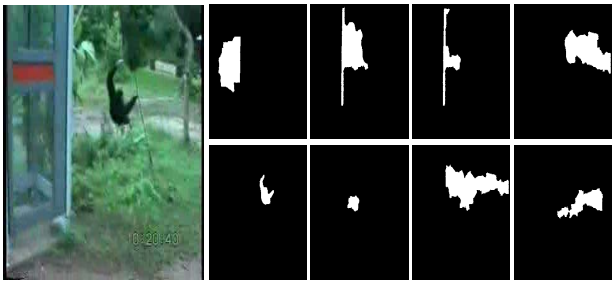

Figure 2. Object proposals from a video frame employing the method in [7]. The left side image is one of the video frames. Note that the monkey is the object of interest in the frame. Images on the right show some of the top ranked object proposals from the frame. Most of the proposals do not correspond to an actual object. The goal of the proposed work is to generate an enhanced set of object proposals and extract the segments related to the primary object from the video.

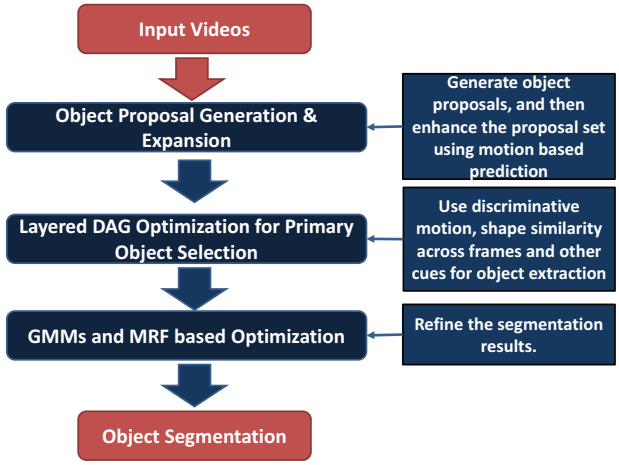

Figure 3. The Video Object Segmentation Framework

tion results. However, most computer vision applications involve processing of large amounts of video data, which makes manual initialization cost prohibitive. Consequently, a large number of automatic methods have also been proposed for video object segmentation. A subset of these methods employs motion grouping ([19, 18, 4]) for object segmentation. Other methods ([10, 3, 21]) use appearance cues to segment each frame first and then use both appearance and motion constraints for a bottom-up final segmentation. Methods like [9, 3, 11, 22] present efficient optimization frameworks for spatiotemporal grouping of pixels for video segmentation. However, all of these automatic methods do not have an explicit model of how an object looks or moves, and therefore, the segments usually don't correspond to a particular object but only to image regions that exhibit coherent appearance or motion.

Recently, several methods ([7, 5, 1]) were proposed that provided an explicit notion of how a generic object looks like. Specifically, the method [7] could extract object-like regions or 'object proposals' from images. This work was built upon by Lee et al. [13] and Ma and Latecki [14] to employ object proposals for object video segmentation. Lee et al. [13] proposed to detect the primary object by collecting a pool of object proposals from the video, and then applying spectral graph clustering to obtain multiple binary inlier/outlier partitions. Each inlier cluster corresponds to a particular object's regions. Both motion and appearance based cues are used to measure the 'objectness' of a proposal in the cluster. The cluster with the largest average 'objectness' is likely to contain the primary object in video. One shortcoming of this approach is that the clustering process ignores the order of the proposals in the video, and therefore, cannot model the evolution of object's shape and location with time. The work by Ma and Latecki [14] attempts to mitigate this issue by utilizing relationships between object proposals in adjacent frames. The object region selection problem is modeled as a constrained Maximum Weight Cliques problem in order to find the true object region from all the video frames simultaneously. However, this problem is NP-hard ([14]) and an approximate optimization technique is used to obtain the solution. The object proposal based segmentation approaches [13, 14] have two additional limitations compared to the proposed method. First, in both approaches, object proposal generation for a particular frame doesn't directly depend on object proposals generated for adjacent frames. Second, both approaches do not actually predict the shape of the object in adjacent frames when computing region similarity, which degrades segmentation performance for fast moving objects.

In this paper, we present an approach that though inspired from aforementioned approaches, attempts to remove their shortcomings. Note that, in general, an object's shape and appearance varies slowly from frame to frame. Therefore, the intuition is that the object proposal sequence in a video with high 'objectness', and high similarity across frames is likely to be the primary object. To this end, we use optical flow to track the evolution of object shape, and compute the difference between predicted and actual shape (along with appearance) to measure similarity of object proposals across frames. The 'objectness' is measured using appearance and a motion based criterion that emphasizes high optical flow gradients at the boundaries between objects proposals and the background. Moreover, the primary object proposal selection problem is formulated as the longest path problem for Directed Acyclic Graph (DAG), for which (unlike [14]) an optimal solution exists in linear time. Note that, if the temporal order of object proposals locations (across frames) is not used ([13], then it can result in no proposals being associated with the prima-

ry object for many frames (please see Figure 1). The proposed method not only uses object proposals from a particular frame (please see Figure 2), but also expands the proposal set using predictions from proposals of neighboring frame. The combination of proposal expansion, and the predicted shape based similarity criteria results in temporally dense and spatially accurate primary object proposal extraction. We have evaluated the proposed approach using several challenging benchmark videos and it outperforms both unsupervised and supervised state-of-the-art methods

In Section 2, the proposed layered DAG based object selection approach is introduced and discussed in detail; In Section 3, both qualitative and quantitative experiments results for two publicly available datasets and some other challenging videos are shown; The paper is concluded in Section 4.

2. Layered DAG based Video Object Segmentation

2.1. The Framework

The proposed framework consists of 3 stages (as shown in Figure 3): **1**. Generation of object proposals per-frame and then expansion of the proposal set for each frame based on object proposals in adjacent frames. **2**. Generation of a layered DAG from all the object proposals in the video. The longest path in the graph fulfills the goal of maximizing objectness and similarity scores, and represents the most likely set of proposals denoting the primary object in the video. **3**. The primary object proposals are used to build object and background models using Gaussian mixtures, and a graph-cuts based optimization method is used to obtain refined per-pixel segmentation. Since the proposed approach is centered around layered DAG framework for selection of primary object regions, we will start with its description.

2.2. Layered DAG Structure

We want to extract object proposals with high objectness likelihood, high appearance similarity and smoothly varying shape from the set of all proposals obtained from the video. Also since we want to extract the primary object only, we want to extract at most a single proposal per frame. Keeping these objectives in mind, the layered DAG is formed as follows. Each object proposal is represented by two nodes: a 'beginning node' and an 'ending node' and there are two types of edges: unary edges and binary edges. The unary edges have weights which measure the objectness of a proposal. The details of the function for unary weight assignments (measuring objectness) are given in section 2.2.1. All the beginning nodes in the same frame form a layer, so as the ending nodes. A directed unary edge is built from beginning node to ending node. Thus, each video frame is represented by two layers in the graph. Directed binary edges are built from any ending node to all the beginning nodes in latter layers. The binary edges have weights which measure the appearance and shape similarity between the corresponding object proposals across frames. The binary weight assignment functions are introduced in Section 2.2.2.

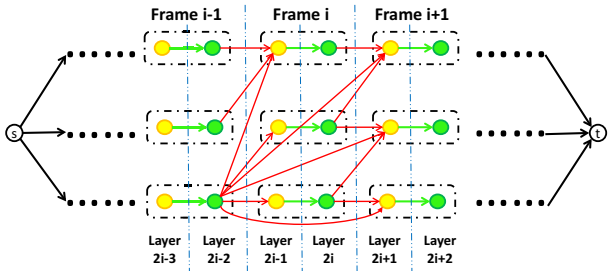

Figure 4. Layered Directed Acyclic Graph (DAG) Structure. Node "s" and "t" are source and sink nodes respectively, which have zero weights for edges with other nodes in the graph. The yellow nodes and the green nodes are "beginning nodes" and "ending nodes" respectively and they are paired such that each yellow-green pair represents an object proposal. All the beginning nodes in the same frame are arranged in a layer and the same as the ending nodes. The green edges are the unary edges and red edges are the binary edges.

Figure 4 is an illustration of the graph structure. It shows frame $i-1$, i and $i+1$ of the graph, with corresponding layers of $2i-3$, $2i-2$, $2i-1$, $2i$, $2i+1$ and $2i+2$. Note that, only 3 object proposals are shown for each layer for simplicity, however, there are usually hundreds of object proposals for each frame and the number of object proposals for different frames are not necessary the same. The yellow nodes are "beginning nodes", the green nodes are "ending nodes", the green edges are unary edges with weights indicating objectness and the red edges are binary edges with weights indicating appearance and shape similarity (note that the graph only shows some of the binary edges for simplicity). There is also a virtual source node s and a sink node t with 0 weighted edges (black edges) to the graph. Note that, it is not necessary to build binary edges from an ending node to all the beginning nodes in latter layers. In practice, only building binary edges to the next three subsequent frames is enough for most of the videos.

2.2.1 Unary Edges

Unary edges measure the objectness of the proposals. Both appearance and motion are important to infer the objectness, so the scoring function for object proposals is defined as $S_{unary}(r) = A(r) + M(r)$, in which r is any object proposal, $A(r)$ is the appearance score and $M(r)$ is the motion score. We define $M(r)$ as the average Frobenius norm of optical flow gradient around the boundary of object pro-

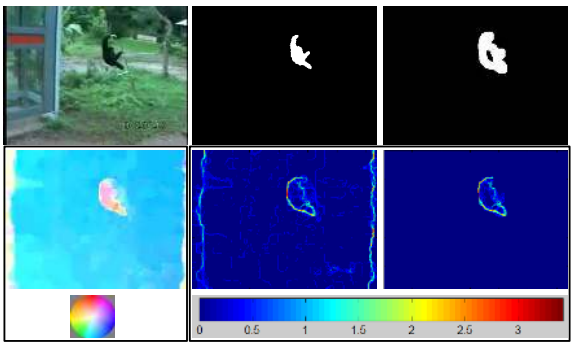

Figure 5. Optical Flow Gradient Magnitude Motion Scoring. In row 1, column 1 shows the original video frame, column 2 is one of the object proposals and column 3 shows dilated boundary of the object proposal. In row 2, column 1 shows the forward optical flow of the frame, column 2 shows the optical flow gradient magnitude map and column 3 shows the optical flow gradient magnitude response for the specific object proposal around the boundary. [Please Print in Color]

posal r. The Frobenius norm of optical flow gradients is defined as:

$$\|U_X\|_F = \left\| \begin{bmatrix} u_x & u_y \\ v_x & v_y \end{bmatrix} \right\|_F = \sqrt{u_x^2 + u_y^2 + v_x^2 + v_y^2}, \tag{1}$$

in which $U = (u, v)$ is the forward optical flow of the frame, u_x, v_x and u_y, v_y are optical flow gradients in x and y directions respectively.

The intuition behind this motion scoring function is that, the motions of foreground object and background are usually distinct, so boundary of moving objects usually implies discontinuity in motion. Therefore, ideally, the gradient of optical flow should have high magnitude around foreground object boundary (this phenomenon could be easily observed from Figure 5). In equation 1, we use the Frobenius norm to measure the optical flow gradient magnitude, the higher the value, the more likely the region is from a moving object. In practice, usually the maximum of optical flow gradient magnitude does not coincide exactly with the moving object boundary due to underlying approximation of optical flow calculation. Therefore, we dilate the object proposal boundary and get the average optical flow gradient magnitude as the motion score. Figure 5 is an illustration of this process. The appearance scoring function $A(r)$ is measured by the objectness ([7]).

2.2.2 Binary Edges

Binary edges measure the similarity between object proposals across frames. For measuring the similarity of regions, color, location, size and shape are the properties to be considered. We define the similarity between regions as the weight of binary edges as follows:

$$S_{binary}(r_m, r_n) = \lambda \cdot S_{color}(r_m, r_n) \cdot S_{overlap}(r_m, r_n), \tag{2}$$

in which r_m and r_n are regions from frame m and n, λ is a constant value for adjusting the ratio between unary and binary edges, $S_{overlap}$ is the overlap similarity between regions and S_{color} is the color histogram similarity:

$$S_{color}(r_m, r_n) = hist(r_m) \cdot hist(r_n)^T, \tag{3}$$

in which $hist(r)$ is the normalized color histogram for a region r.

$$S_{overlap}(r_m, r_n) = \frac{|r_m \cap warp_{mn}(r_n)|}{|r_m \cup warp_{mn}(r_n)|}, \tag{4}$$

in which $warp_{mn}(r_n)$ is the warped region from r_n by optical flow to frame m. It is clear that S_{color} encodes the color similarity between regions and $S_{overlap}$ encodes the size and location similarity between regions. If two regions are close, and the sizes and shapes are similar, the value would be higher, and vice versa. Note that, unlike prior approaches [13, 14], we use optical flow to predict the region (i.e. encoding location and shape), and therefore we are better able to compute similarity for fast moving objects.

2.2.3 Dynamic Programming Solution

Until now, we have built the layered DAG and the objective is clear: to find the highest weighted path in the DAG. Assume the graph contains $2F + 2$ layers (F is the frame number), the source node is in layer 0 and the sink node is in layer $2F + 2$. Let N_{ij} denotes the jth node in ith layer and $E(N_{ij}, N_{kl})$ denotes the edge from N_{ij} to N_{kl}. Layer i has M_i nodes. Let $P = (p_1, p_2, ..., p_{m+1}) = (N_{01}, N_{j_1 j_2}, ..., N_{j_{m-1} j_m}, N_{(2n+2)1})$ be a path from source to sink node. Therefore,

$$P_{max} = arg \max_P \sum_{i=1}^{m} E(p_i, p_{i+1}). \tag{5}$$

P_{max} forms a Longest (simple) Path Problem for DAG. Let $OPT(i, j)$ be the maximum path value for N_{ij} from source node. The maximum path value satisfies the following recurrence for $i \geq 1$ and $j \geq 1$:

$$OPT(i, j) = \max_{k=0...i-1, l=1...M_k} [OPT(k, l) + E(N_{kl}, N_{ij})]. \tag{6}$$

This problem could be solved by dynamic programming in linear time [12]. The computational complexity for the algorithm is $O(n + m)$, in which n is the number of nodes

and m is the number of edges. The most important parameter for the layered DAG is the ratio λ between unary edges and binary edges. However, in practice, the results are not sensitive to it, and in the experiments λ is simply set to be 1.

2.3. Per-pixel Video Object Segmentation

Once the primary object proposals are obtained in a video, the results are further refined by a graph-based method to get per-pixel segmentation results. We define a spatiotemporal graph by connecting frames temporally with optical flow displacement. Each of the nodes in the graph is a pixel in a frame, and edges are set to be the 8-neighbors within one frame and the forward-backward 18 neighbors in adjacent frames. We define the energy function for labeling $f = [f_1, f_2, ..., f_n]$ of n pixels with prior knowledge of h:

$$E(f,h) = \sum_{i \in S} D_i^h(f_i) + \lambda \sum_{(i,j) \in N} V_{i,j}(f_i, f_j), \quad (7)$$

where $S = \{p_i, ..., p_n\}$ is the set of n pixels in the video, N consists of neighboring pixels, and i,j index the pixels. p_i could be set to 0 or 1 which represents background or foreground respectively. The unary term D_i^h defines the cost of labeling pixel i with label f_i which we get from the Gaussian Mixture Models (GMM) for both color and location.

$$D_i^h(f_i) = -log(\alpha U_i^c(f_i, h) + (1-\alpha) U_i^l(f_i, h)), \quad (8)$$

where $U_i^c(.)$ is the color-induced cost and $U_i^l(.)$ is the location cost.

For the binary term $V_{i,j}(f_i, f_j)$, we follow the definitions in [17]:

$$V_{i,j}(f_i, f_j) = [f_i \neq f_j] exp^{-\beta(C_i - C_j)^2}, \quad (9)$$

where $[.]$ denotes the indicator function taking values 0 and 1, $(C_i - C_j)^2$ is the Euclidean distance between two adjacent nodes in RGB space, and $\beta = (2\sum (C_i - C_j)^2)^{-1}|_{(i,j) \in N}$

We use the graph-cuts based minimization method in [8] to obtain the optimal solution for equation 7, and thus get the final segmentation results. Next, we describe the method for object proposal generation that is used to initialize the video object segmentation process.

2.4. Object Proposal Generation & Expansion

In order to achieve our goal of identifying image regions belonging to the primary object in the video, it is preferable (though not necessary) to have an object proposal corresponding to the actual object for each frame in which object is present. Using only appearance or optical flow based cues to generate object proposals is usually not enough for this purpose. This phenomenon could be observed in the example shown in Figure 6. For frame i in this figure, hundreds of object proposals were generated using method in [7], however, no proposal is consistent with the true object, and the object is fragmented between different proposals.

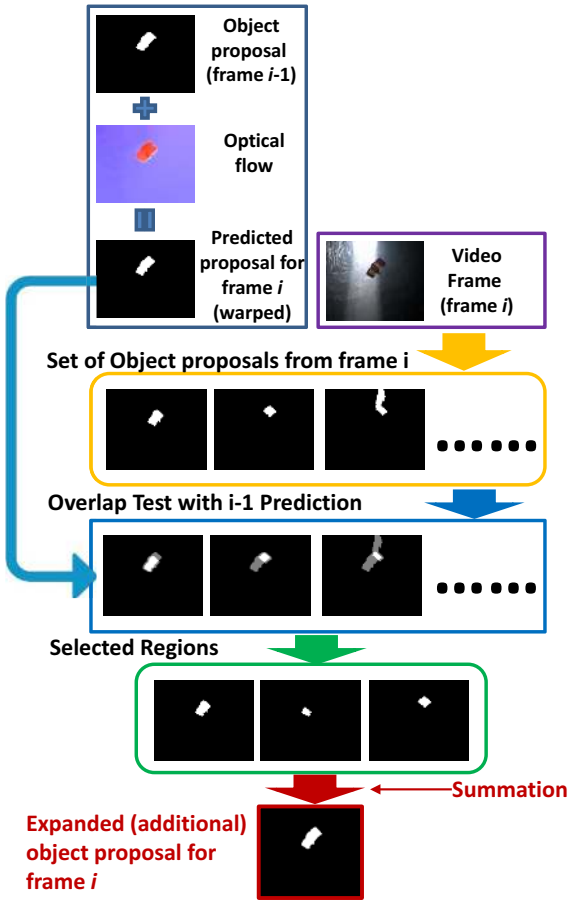

Figure 6. Object Proposal Expansion. For each optical flow warped object proposal in frame $i-1$, we look for object proposals in frame i which have high overlap ratios with the warped one. If some object proposals all have high overlap ratios with the warped one, they are merged into a new large object proposal. This process will produce the right object proposal if it is not discovered by [7] from frame i, but frame $i-1$.

We assume that an object's shape and location changes smoothly across frames and propose to enhance the set of object proposals for a frame by using the proposals generated for its adjacent frames. The object proposal expansion method works by the guidance of optical flow (see Figure 6). For the forward version of object proposal expansion, each object proposal r_{i-1}^k in frame $i-1$ is warped by the forward optical flow to frame i, then a check is made if any proposal r_i^j in frame i has a large overlap ratio with the

warped object proposal, i.e.,

$$o = \frac{|warp_{i-1,i}(r_{i-1}^k) \cap r_i^j|}{|r_i^j|}. \quad (10)$$

The contiguous overlapped areas, for regions in i+1 with o greater than 0.5, are merged into a single region, and are used as additional proposals. Note that, the old original proposals are also kept, so this is an 'expansion' of the proposal set, and not a replacement. In practice, this process is carried out both forward and backward in time. Since it is an iterative process, even if suitable object proposals are missing in consecutive frames, they could potentially be produced by this expansion process. Figure 6 shows an example image sequence where the expansion process resulted in generation of a suitable proposal.

3. Experiments

The proposed method was evaluated using two well-known segmentation datasets: SegTrack dataset [20] and GaTech video segmentation dataset [9]. Quantitative comparisons are shown for SegTrack dataset since ground-truth is available for this dataset. Qualitative results are shown for GaTech video segmentation dataset. We also evaluated the proposed approach on additional challenging videos, for which we will share the ground-truth to aid future evaluations.

3.1. SegTrack Dataset

We first evaluate our method on Segtrack dataset [20]. There are 6 videos in this dataset, and also a pixel-level segmentation ground-truth for each video is available. We follow the setup in the literature ([13, 14]), and use 5 (birdfall, cheetah, girl, monkeydog and parachute) of the videos for evaluation (since the ground-truth for the other one (penguin) is not useable). We use an optical flow magnitude based model selection method to infer the camera motion: for static cameras, a background subtraction cue is also used for moving object extraction; for all the results shown in this section, the static camera model was only selected (automatically) for the "birdfall" video.

We compare our method with 4 state-of-the-art methods [14], [13], [20] and [6] shown in Table 1. Note that our method is a unsupervised method, and it outperforms all the other unsupervised methods except for the parachute video where it is a close second. Note that [20] and [6] are supervised methods which need an initial annotation for the first frame. The results in Table 1 are the average per-frame pixel error rate compared to the ground-truth. The definition is [20]:

$$error = \frac{XOR(f, GT)}{F}, \quad (11)$$

where f is the segmentation labeling results of the method, GT is the ground-truth labeling of the video, and F is the number of frames in the video. Figure 7 shows qualitative results for the videos of SegTrack dataset.

Figure 7. SegTrack dataset results. The regions within the red boundaries are the segmented primary objects. [Please Print in Color]

Video	Ours	[14]	[13]	[20]	[6]
birdfall	**155**	189	288	252	454
cheetah	**633**	806	905	1142	1217
girl	**1488**	1698	1785	1304	1755
monkeydog	**365**	472	521	563	683
parachute	220	221	**201**	235	502
Avg.	**452**	542	592	594	791
supervised?	N	N	N	Y	Y

Table 1. Quantitative results and comparison with the state of the art on SegTrack dataset

Figure 8 is an example that shows the effectiveness of the proposed layered DAG approach for temporally dense extraction of primary object regions. The figure shows consecutive frames (frame 38 to frame 43) from "monkeydog" video. The top 2 rows show the results of key-frame object extraction method [13], and the bottom 2 rows show our object region selection results. As one can see, [13] detects the primary object proposal in only one of the frames, however, by using the proposed approach, we can extract the

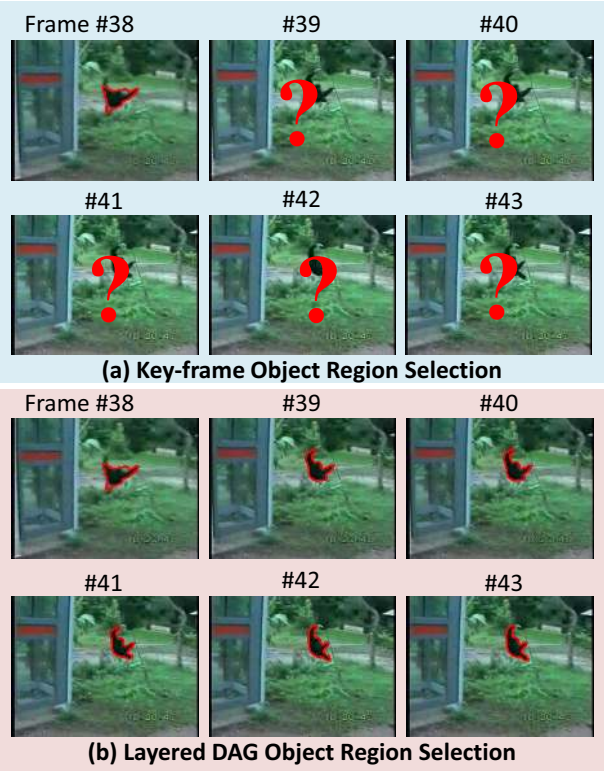

Figure 8. Comparison of object region selection methods. The regions within the red boundaries are the selected object regions. "?" means there is no object region selected by the method. Numbers above are the frame indices.[Please Print in Color]

primary object region from all the frames. This is the main reason that the segmentation results of the proposed method are better than prior methods.

3.2. GaTech Segmentation Dataset

We also evaluated the proposed method on GaTech video segmentation dataset. We show qualitative comparison of results between the proposed approach and the original bottom-up method for the dataset in Figure 9. As one can observe, our results could segment the true foreground object from the background. The method [9] doesn't use an object model which induces over-segmentation (although the results are very good for the general segmentation problem).

3.3. Persons and Cars Segmentation Dataset

We have built a new dataset for video object segmentation. The dataset is challenging: persons are in a variety of poses; cars have different speeds, and when they are slow, it is very hard to do motion segmentation. We generate ground truth for those videos. Figure 10 shows some sample results from this dataset, and Table 2 shows the quantitative

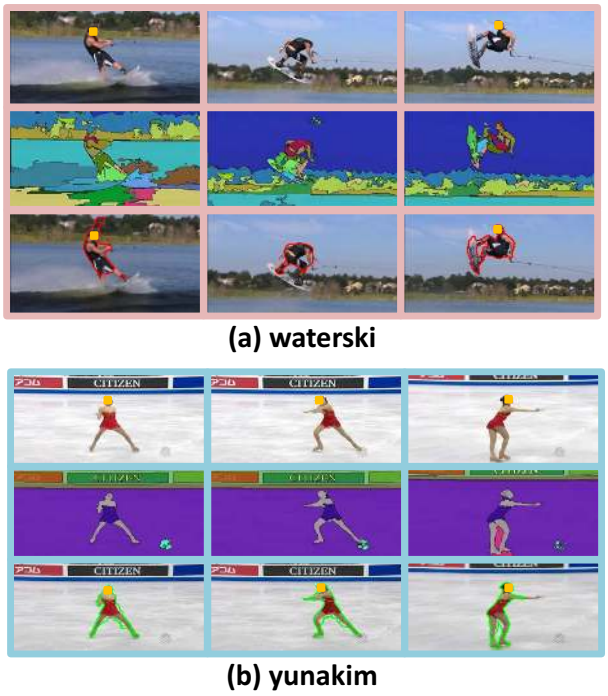

Figure 9. Object Segmentation Results on GaTech Video Segmentation Dataset. Row 1: orignial frame, Row 2: Segmentation results by the bottom-up segmentation method [9]. Row 3: Video object segmentation by the proposed method. The regions within the red or green boundaries are the segmented primary objects. [Please Print in Color]

Video	Average per-frame pixel error
Surfing	**1209**
Jumping	**835**
Skiing	**817**
Sliding	**2228**
Big car	**1129**
Small car	**272**

Table 2. Quantitative Results on Persons and Cars dataset

results for this dataset (the average per-frame pixel error is defined as the same as SegTrack dataset [20]). Please go to http://crcv.ucf.edu for more details.

4. Conclusions

We have proposed a novel and efficient layered DAG based approach to segment the primary object in videos. This approach also uses innovative mechanisms to compute the 'objectness' of a region and to compute similarity between object proposals across frames. The proposed approach outperforms the state of the art on the well-known SegTrack dataset. We also demonstrate good segmentation performance on additional challenging data sets.

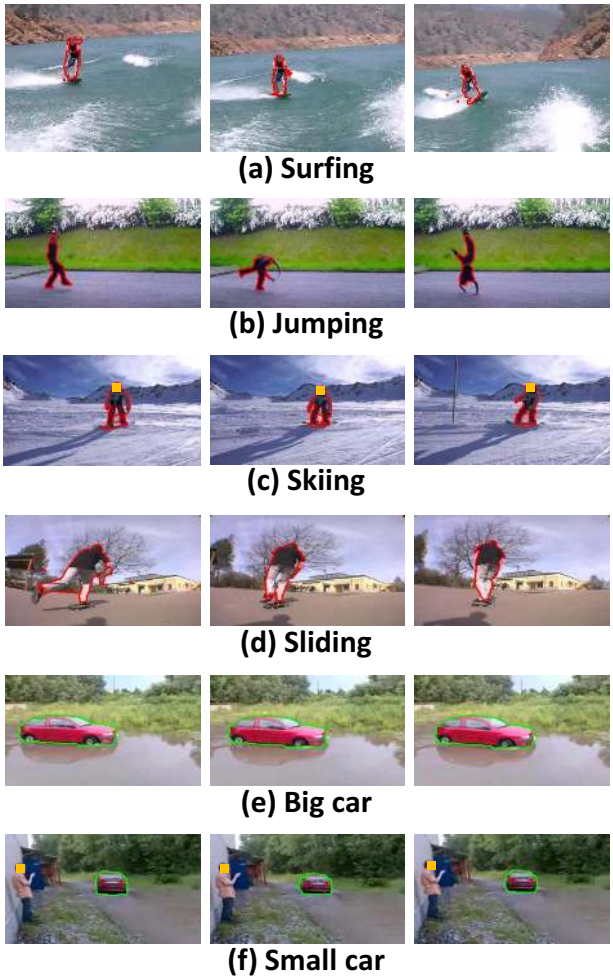

Figure 10. Sample Results on Persons and Cars Dataset. Please go to http://crcv.ucf.edu for more details.

Acknowledgment

This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract numbers D11PC20066. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S.Government.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *CVPR*, pages 73–80, 2010.

[2] X. Bai, J. Wang, D. Simons, and G. Sapiro. Video snapcut: robust video object cutout using localized classifiers. *ACM Transactions on Graphics*, 28(3):70, 2009.

[3] W. Brendel and S. Todorovic. Video object segmentation by tracking regions. In *ICCV*, pages 833–840, 2009.

[4] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, pages 282–295, 2010.

[5] J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *CVPR*, pages 3241–3248, 2010.

[6] P. Chockalingam, N. Pradeep, and S. Birchfield. Adaptive fragments-based tracking of non-rigid objects using level sets. In *ICCV*, pages 1530–1537, 2009.

[7] I. Endres and D. Hoiem. Category independent object proposals. In *ECCV*, pages 575–588, 2010.

[8] B. Fulkerson, A. Vedaldi, and S. Soatto. Class segmentation and object localization with superpixel neighborhoods. In *ICCV*, pages 670–677, 2009.

[9] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *CVPR*, pages 2141–2148, 2010.

[10] Y. Huang, Q. Liu, and D. Metaxas. Video object segmentation by hypergraph cut. In *CVPR*, pages 1738–1745, 2009.

[11] J.Wang, B. Thiesson, Y. Xu, and M. Cohen. Image and video segmentation by anisotropic kernel mean shift. In *ECCV*, 2004.

[12] J. Kleinberg and E. Tardos. *Algorithm design*. Pearson Education and Addison Wesley, 2006.

[13] Y. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *ICCV*, pages 1995–2002, 2011.

[14] T. Ma and L. Latecki. Maximum weight cliques with mutex constraints for video object segmentation. In *CVPR*, pages 670–677, 2012.

[15] B. Price, B. Morse, and S. Cohen. Livecut: Learning-based interactive video segmentation by evaluation of multiple propagated cues. In *ICCV*, pages 779–786, 2009.

[16] X. Ren and J. Malik. Tracking as repeated figure/ground segmentation. In *CVPR*, pages 1–8, 2007.

[17] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM Transactions on Graphics*, volume 23, pages 309–314, 2004.

[18] Y. Sheikh, O. Javed, and T. Kanade. Background subtraction for freely moving cameras. In *ICCV*, pages 1219–1225, 2009.

[19] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *ICCV*, pages 1154–1160, 1998.

[20] D. Tsai, M. Flagg, and J. Rehg. Motion coherent tracking with multi-label mrf optimization. In *BMVC*, page 1, 2010.

[21] A. Vazquez-Reina, S. Avidan, H. Pfister, and E. Miller. Multiple hypothesis video segmentation from superpixel flows. In *ECCV*, pages 268–281, 2010.

[22] C. Xu, C. Xiong, and J. J. Corso. Streaming hierarchical video segmentation. In *ECCV*, pages 626–639. 2012.

[23] J. Yuen, B. Russell, C. Liu, and A. Torralba. Labelme video: Building a video database with human annotations. In *ICCV*, pages 1451–1458, 2009.