Video Segmentation via Object Flow

Yi-Hsuan Tsai
UC Merced
ytsai2@ucmerced.edu

Ming-Hsuan Yang
UC Merced
mhyang@ucmerced.edu

Michael J. Black
MPI for Intelligent Systems
black@tuebingen.mpg.de

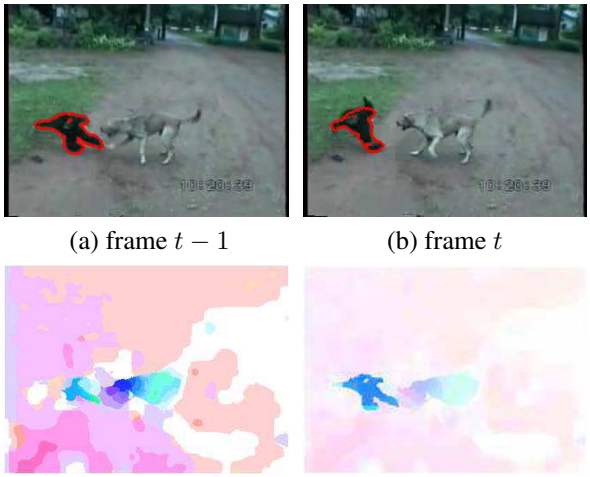
Abstract

Video object segmentation is challenging due to fast moving objects, deforming shapes, and cluttered backgrounds. Optical flow can be used to propagate an object segmentation over time but, unfortunately, flow is often inaccurate, particularly around object boundaries. Such boundaries are precisely where we want our segmentation to be accurate. To obtain accurate segmentation across time, we propose an efficient algorithm that considers video segmentation and optical flow estimation simultaneously. For video segmentation, we formulate a principled, multi-scale, spatio-temporal objective function that uses optical flow to propagate information between frames. For optical flow estimation, particularly at object boundaries, we compute the flow independently in the segmented regions and recompose the results. We call the process object flow *and demonstrate the effectiveness of jointly optimizing optical flow and video segmentation using an iterative scheme. Experiments on the SegTrack v2 and Youtube-Objects datasets show that the proposed algorithm performs favorably against the other state-of-the-art methods.*

1. Introduction

Our goal is to segment video sequences, classifying each pixel as corresponding to a foreground object or the background in every frame. Critical to solving this task is the integration of information over time to maintain a consistent segmentation across the entire video. Numerous methods have been proposed to enforce temporal consistency in videos by tracking pixels, superpixels or object proposals [5, 10, 12, 19, 26, 35, 38, 40]. Another line of research formulates this problem with a graphical model and propagates the foreground regions throughout an image sequence [1, 11, 14, 35, 36]. In addition, several algorithms [18, 19, 21, 24, 45] focus on object-level segmentations that favor temporal consistency. Such object-level methods, however, may not be accurate on the pixel level, generating inaccurate object boundaries.

Optical flow estimation has been extensively studied [2, 6, 30] and it is widely used for video segmentation and related problems [8, 13, 23, 42]. For instance, graph-based video segmentation methods [14, 35] use optical flow in the formulation of pairwise potentials that ensure frame-to-frame segmentation consistency. However, estimated optical flow may contain significant errors, particularly due to large displacements or occlusions [6, 9, 29, 39]. To compute accurate optical flow fields, it is common to segment images or extract edges to preserve motion details around object boundaries [3, 43, 44, 47]. However, most methods do not consider both flow estimation and video segmentation together.



(a) frame $t-1$ (b) frame t

(c) initial optical flow (d) updated optical flow

Figure 1. **Object flow.** (a)-(b) two consecutive frames. (c) optical flow computed by [30] from frame $t-1$ to t. (d) optical flow that is updated using the segmentation marked by the red contour. The motions within the object are more consistent and the motion boundaries are more precise compared with the initial flow.

In contrast, we estimate object segmentation and optical flow synergistically such that the combination improves both. Figure 1 summarizes the main ideas of this work. If the segmentation of the object is known, optical flow within the same object should be smooth but flow across the boundary need not be smooth. Similarly if an object moves differently from the background, then the motion boundary will correspond to the object boundary. Hence, accurate optical flow facilitates detecting precise object boundaries and

vice versa.

This notion, of course, is not entirely new, but few methods have tried to integrate video segmentation and flow estimation. Specifically, in this paper, we address the above problems by considering object segmentation and optical flow simultaneously, and propose an efficient algorithm, which we refer as *object flow*. For the segmentation model, we construct a multi-level graphical model that consists of pixels and superpixels, where each of these play different roles for segmentation. On the superpixel level, each superpixel is likely to contain pixels from the foreground and background as the object boundary may not be clear. On the pixel level, each pixel is less informative although it can be used for more accurate estimation of motion and segmentation. With the combination of these two levels, the details around the object boundary can be better identified by exploiting both statistics contained in superpixels and details on the pixel level. Furthermore, we generate superpixels by utilizing supervoxels [13, 42] between two frames to exploit temporal information in addition to the use of optical flow. After obtaining the segmentation results, we apply the foreground and background information to re-estimate optical flow (Figure 1), and then iteratively use the updated optical flow to re-segment the object region.

We evaluate the proposed object flow algorithm on the SegTrack v2 [19] and Youtube-Objects [25] datasets. We work in the standard paradigm of tracking that assumes an initialization of the object segmentation in the first frame, which could come from simple user input [27] . We quantitatively compare our segmentation accuracy to other state-of-the-art results and show improvements to the estimated optical flow. With the updated optical flow using the segmentation, we show that the iterative procedure improves both segmentation and optical flow results in terms of visual quality and accuracy.

The contributions of this work are as follows. First, we propose a multi-level spatial-temporal graphical model for video object segmentation and demonstrate that it performs better than single-level models. Second, we show that the segmentation results can be used to refine the optical flow, and vice versa, in the proposed object flow algorithm. Third, we demonstrate that our joint model of segmentation and optical flow can be efficiently computed by iterative optimization.

2. Related Work and Problem Context

Segment-based Tracking. Several segment-based tracking methods [12, 19, 38, 40] have been developed. A Hough voting based algorithm [12] performs online tracking and generates segmentation using GrabCut [27]. Wang *et al*. [38] propose a discriminative appearance model based on superpixels to distinguish the target object from the background. A recent part-based method tracks object segments [40] with the assumption that the number of parts and superpixels are known in advance. As these approaches only consider superpixels that are generated independently in each frame, they do not explicitly exploit temporal information among regions.

Li *et al*. [19] track object-level region proposals in consecutive frames. However, it is computationally expensive to compute region proposals and the generated object boundaries are not accurate. Lalos *et al*. [17] also propose an algorithm called object flow but the goals are significantly different from ours in that they focus on estimating displacements of objects. In addition, this method neither addresses generic flow estimation nor integrates segmentation and flow.

Graph-based Models. One approach to segment objects in videos is to propagate foreground labels [1, 14, 35, 36] between frames based on graphical models. Graphical models for video segmentation typically use unary terms that are determined by the foreground appearance, motions or locations, and pairwise terms that encode spatial and temporal smoothnesses. These methods typically use optical flow to maintain temporal links, but they are likely to fail when the flow is inaccurate. In addition, graphical models can be used for refining segments [18, 24]. Lee *et al*. [18] use ranked object proposals and select key segments for shape matching. Similar to the issues mentioned in [19], it is computationally expensive to generate proposals and they are likely to contain foreground and background pixels.

Layered Models. Video segmentation and motion estimation are closely related. Layered models [7, 15, 16, 34, 37, 46] jointly optimize for segmentation and optical flow. These models can be extended to the temporal domain with more than two frames [31]. Early methods focus only on motion information but more recent formulations combine image and motion information in segmenting the scene into layers [32, 41]. Most layered methods use complicated and computationally expensive inference, thereby limiting their applications.

In this paper, we propose an efficient algorithm that jointly updates object segmentation and optical flow models using an iterative scheme. The optical flow helps identify temporal connections throughout the video, and the segmentation improves the optical flow estimation at motion boundaries.

3. Video Object Segmentation

In this section, we first explain how we construct the object segmentation model. Given the initialization in the first frame, we aim to propagate the foreground label throughout the entire video. Note that, in contrast to unsupervised methods [18, 19], that rely on motion and object proposals and process the entire video offline in batch mode,

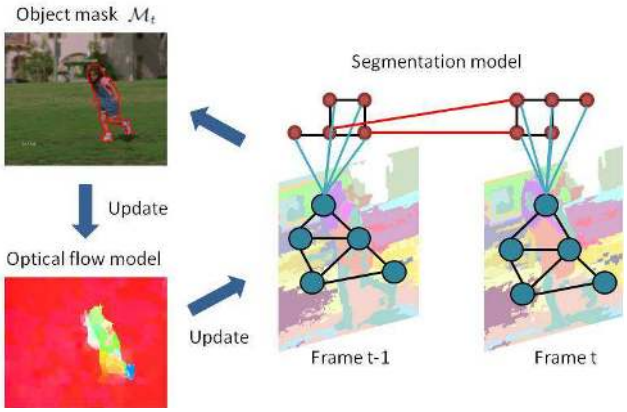


Figure 2. Overview of the proposed model. For segmentation, we consider a multi-level spatial-temporal model. Red circles denote pixels, which belong to the superpixel marked by the turquoise circles. The black and the red lines denote the spatial and temporal relationships, respectively. The relationships between the pixels and the superpixel are denoted by the turquoise lines. After obtaining the object mask, $\mathcal{M}_t$, we use this mask to re-estimate the optical flow, and update both models iteratively.

the proposed algorithm is able to track and segment objects for online applications. Before assigning each pixel a label, we search the possible locations for the object in each frame to reduce the background noise. A multi-level spatial-temporal graphical model is then applied to the estimated object regions. In this stage, unary and pairwise terms for superpixels are introduced by the supervoxel to better ensure temporal consistency. Figure 2 shows the proposed multi-level segmentation model.

Object Location. Instead of using the segmentation model on the entire image, we first estimate the object location to reduce the computational load and the effect of background noise. We design a scoring function for each pixel based on color and location:

$$S_t(x_i^t) = A_t(x_i^t) + L_t(x_i^t, \mathcal{M}_{t-1}), \tag{1}$$

where A_t is the color score on the pixel x_i^t computed by a Gaussian Mixture Model (GMM), and L_t is the location score measured by the Euclidean distance transform of the binary object mask $\mathcal{M}_{t-1}$ in the previous frame.

Since we do not know the exact object location or shape in the current frame, we assume that the object does not move abruptly. Therefore, we generate the rough object mask in the current frame t using the segmentation mask $\mathcal{M}_{t-1}$ in the previous frame translated by the average optical flow vector within the mask. We then use and expand this coarse mask on a local search region that is s times the size of the object mask $\mathcal{M}_{t-1}$ (s is from 2 to 3 depending on the object size in this work). This mask ensures that most of the pixels within the object are covered. After obtaining the local search region, we use the distance transform on

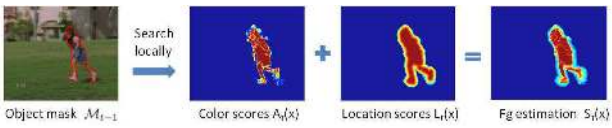


Figure 3. Estimated object location R_t. Given the object mask $\mathcal{M}_{t-1}$ marked as the red contour, we search for a local region in the current frame t and compute the foreground scores based on color and location. The estimated foreground region is used for label assignment.

this expanded mask to compute location scores. Similarly, we also consider color scores based on this local region. Figure 3 illustrates one example of the combination of two scores. A threshold is then applied to select the object location R_t for further determining label assignment.

Graphical Model. After the object region for label assignment is selected, we utilize a spatial-temporal graphical model to assign each pixel with a foreground or background label. We define an energy function in a Conditional Random Field (CRF) form for the pixel $x_i^t \in X$ with label $\in \{0, 1\}$:

$$E_{pix}(X) = \bar{U}_t(X, \mathcal{M}_{t-1}) + \gamma_1^s \sum_{(i,j,t)\in\mathcal{E}_t} \bar{V}_t(x_i^t, x_j^t) + \gamma_1^t \sum_{(i,j,t)\in\mathcal{E}_t} \bar{W}_t(x_i^{t-1}, x_j^t), \tag{2}$$

where $\bar{U}_t$ is the unary potential for the cost to be foreground or background, and $\bar{V}_t$ and $\bar{W}_t$ are pairwise potentials for spatial and temporal smoothnesses with weights γ_1^s and γ_1^t, respectively. Both of the pairwise terms are defined as in [24]. Note that we only consider $\mathcal{E}_t$ within the region R_t generated in the object location estimation step.

For the unary term in (2), we consider appearance and location energies defined by $\bar{U}_t(X, \mathcal{M}_{t-1})=$

$$\alpha_1 \sum_{(i,t)\in R_t} \bar{\Phi}_a(x_i^t) + \beta_1 \sum_{(i,t)\in R_t} \bar{\Phi}_l(x_i^t, \mathcal{M}_{t-1}), \tag{3}$$

where $\bar{\Phi}_a$ is the appearance term, and $\bar{\Phi}_l$ is the location term defined similar to the one in (1). The difference is that for the nodes in the previous frame, we can simply compute the distance transform of the object mask $\mathcal{M}_{t-1}$. For the appearance term, we construct the color GMM in the first frame, and an online SVM model with CNN features [20] updated every frame. The weight α_1 consists of α_1^{col} and α_1^{cnn} for the color and CNN features, respectively. By minimizing (2), we obtain labels within R_t and thus the object mask $\mathcal{M}_t$, and then continue to propagate to the next frame.

Multi-level Model. Although the model based on pixels can achieve fine-grained segmentation, pixels are usually sensitive to noise when optical flow is not accurately estimated. On the other hand, an alternative way is to use

larger segments such as superpixels that contain more information by considering every pixel in the neighborhood (i.e., spatial support). However, superpixels may not contain the entire object or may have imprecise object boundaries due to occlusion or motion blur (See Figure 4). Therefore, we construct a multi-level graphical model including pixels and superpixels to ensure boundary as well as temporal consistency.

In this model, the energy terms for both pixels and superpixels have unary and pairwise potentials, and a pairwise term is used for the connection where pixels belong to a superpixel (See Figure 2). In addition, since optical flow may not be estimated correctly due to large displacement, we use supervoxels [13] between two frames to generate coherent superpixels and enhance temporal consistency.

The multi-level model is formulated by

$$E_{seg} = \lambda_1 E_{pix}(X) + \lambda_2 E_{sup}(Y) + \lambda_3 E_{pair}(X, Y), \quad (4)$$

where $E_{pix}(X)$ is the model based on pixels in (2); $E_{sup}(Y)$ is the energy function based on superpixels $y_m^t \in Y$; $E_{pair}(X, Y)$ is the pairwise term between pixels and superpixels; and λ_i is the weight. We define $E_{sup}(Y)$ as:

$$E_{sup}(Y) = \hat{U}_t(Y) + \gamma_2 \sum_{(m,n,t)\in\mathcal{E}_t} \hat{V}_t(y_m^t, y_n^t), \quad (5)$$

where $\hat{U}_t$ is the unary potential for labeling a superpixel as foreground or background, and $\hat{V}_t$ is the spatial smoothness within the region R_t. Note that it is not necessary to model the temporal smoothness in (5) since we design a term for the supervoxel and optical flow in the unary term (explained in detail below). The unary term $\hat{U}_t$ is defined in a way similar to (3):

$$\hat{U}_t(Y) = \alpha_2 \sum_{(m,t)\in R_t} \hat{\Phi}_a(y_m^t) + \beta_2 \sum_{(m,t)\in R_t} \hat{\Phi}_l(y_m^t), \quad (6)$$

where $\hat{\Phi}_a$ is the color term defined as the mean color likelihood over all pixels within the superpixel, and a location term $\hat{\Phi}_l$ measures the consistency between the optical flow and the supervoxels. The location term is defined as:

$$\hat{\Phi}_l(y) = flow(y) \times obj(y), \quad (7)$$

where $flow(y)$ is defined by the percentage of pixels in y that are successfully transferred to the next time instant. A successful transfer means that a pixel x_i^{t-1} transfers from a superpixel y_m^{t-1} to a superpixel y_n^t via optical flow, and y_m^{t-1} and y_n^t belong to the same supervoxel. In addition, $obj(y)$ computes the percentage of pixels within the superpixel belonging to the segmentation mask $\mathcal{M}$.

The value of the first term in (7) is high if the supervoxel and optical flow mostly agree with each other. The second term basically measures the likelihood that a superpixel is part of the object. Note that, to compute $obj(y)$ for superpixel nodes in the current frame t, since the object location is unknown, we use the approximate object mask as described in the step for estimating the object location.

pixel: 75.8 superpixel: 82.9 multi-level: 85.5

Figure 4. Segmentation results at different levels with overlap ratios with respect to the ground truth mask. On the pixel or superpixel level, both results are not complete. The results on the pixel level miss part of the leg, while the results on the superpixel level include part of the bike. The multi-level segmentation approach exploits results from both levels for higher accuracy.

$E_{pair}(X, Y)$ models the relationship between superpixels and pixels, encouraging pixels inside the superpixel to have the same label. This pairwise term is defined by

$$E_{pair}(x_i^t, y_m^t) = \begin{cases} 1 - |(p(x_i^t) - p(y_m^t))| & \text{if } l_x \neq l_y \\ 0 & \text{else,} \end{cases}$$

where p is the foreground color probability computed by a color GMM, and l_x and l_y are the labels for the pixel and superpixel. This energy computes the penalty of assigning different labels to pixel x and superpixel y. The subtraction of probabilities indicates how similar x and y are, and the absolute value is from 0 to 1. That is, if the color of the pixel is similar to the mean color of the superpixel, it is likely to have the same label and should have a higher penalty if assigning to different labels.

Overall, to propagate foreground labels, we estimate the object location guided by the optical flow, and utilize a multi-level model to assign the label to each pixel. On the pixel level, optical flow is used to maintain temporal smoothness, whereas for the superpixel, the model measures the consistency between supervoxels and optical flow, and propagates the location information to the next frame.

4. Object Flow

In the previous section, we address video segmentation by utilizing a multi-level spatial-temporal graphical model with the use of optical flow and supervoxels. The ensuing question is how to use the segmentation results to help the estimation of optical flow and vice versa? Since flow vectors within the same object are likely to be similar, we propose to estimate them on the object level. The updated optical flow can then be used again to improve object segmentation. The problem is formulated jointly as follows.

Algorithm 1 Object Flow

Initialize: $\mathbf{u}, \mathbf{v}$ by minimizing (8)
while not converged **do**
  $\mathbf{u}^p \leftarrow \mathbf{u}, \mathbf{v}^p \leftarrow \mathbf{v}$
  Segmentation
    Compute energy terms for (4) using $\mathbf{u}^p, \mathbf{v}^p$
    minimize (4) by graph cuts and obtain $\mathcal{M}$
    $\mathcal{M}^p \leftarrow \mathcal{M}$
  Optical Flow
  if large displacement **then**
      minimize (9) using $\mathcal{M}^p$ and obtain $\mathbf{u}, \mathbf{v}$
      $\mathbf{u}^p \leftarrow \mathbf{u}, \mathbf{v}^p \leftarrow \mathbf{v}$
  end if
  $\mathcal{M} \leftarrow \mathcal{M}^p, \mathbf{u} \leftarrow \mathbf{u}^p, \mathbf{v} \leftarrow \mathbf{v}^p$
end while

Optical Flow with Segmentation. We minimize the classical robust optical flow objective function [30],

$$E(\mathbf{u},\mathbf{v};R) = \sum_{i,j\in R} \{\rho_D(I_{t-1}(i,j) - I_t(i+u_{i,j}, j+v_{i,j})) + \lambda[\rho_S(u_{i,j}-u_{i+1,j}) + \rho_S(u_{i,j}-u_{i,j+1}) + \rho_S(v_{i,j}-v_{i+1,j}) + \rho_S(v_{i,j}-v_{i+1,j})]\}, \quad (8)$$

where $\mathbf{u}$ and $\mathbf{v}$ are the horizontal and vertical components of the optical flow from image I_{t-1} to I_t, and ρ_D and ρ_S are robust data and spatial penalty functions.

We further consider the object mask $\mathcal{M}$ obtained from the segmentation step. One can consider this as a binary assignment of pixels to layers in a layered flow model. Here we use it to decompose the flow problem into two separate estimations,

$$E_{flow}(\mathbf{u}_{fg}, \mathbf{v}_{fg}, \mathbf{u}_{bg}, \mathbf{v}_{bg}) = E(\mathbf{u}_{fg}, \mathbf{v}_{fg}; fg) + E(\mathbf{u}_{bg}, \mathbf{v}_{bg}; bg), \quad (9)$$

where fg and bg are local regions that are slightly larger than the foreground and background regions. This step ensures that the optical flow estimation is less affected by the background noise but still takes partial background regions into account. The final optical flow can be merged by applying the segmentation mask. That is, $\mathbf{u} = \mathcal{M} \cdot \mathbf{u}_{fg} + (1-\mathcal{M}) \cdot \mathbf{u}_{bg}$, which are obtained from $E(\mathbf{u}_{fg}, \mathbf{v}_{fg}; fg)$ and $E(\mathbf{u}_{bg}, \mathbf{v}_{bg}; bg)$, respectively.

Joint Formulation. To formulate the joint problem for segmentation and optical flow, we combine (4) and (9) as:

$$\min_{\mathcal{M},\mathbf{u},\mathbf{v}} E_{total} = E_{seg} + E_{flow}. \quad (10)$$

Note that in E_{seg}, we use optical flow in (2) and (5), and the estimated object location. In E_{flow}, we use the segmentation mask obtained by E_{seg} to decide the foreground and background local regions.

We optimize (10) by iteratively updating the two models once the segmentation or optical flow energy converges. First, we initialize and fix the optical flow to minimize E_{seg} using graph cuts [4]. We then optimize the optical flow by fixing the segmentation mask $\mathcal{M}$, and minimizing E_{flow} using the Classic+NL method [30].

Optimization Details. The main steps of the optimization procedure for (10) are summarized in Algorithm 1. We make a few assumptions to expedite the process. First, we find that for many frames, optical flow can be obtained accurately without the need to re-estimate. For instance, this is true when the object is stationary or moves slightly. In addition, we observe that if the consistency between supervoxels and optical flow is low, it is a good indication that the object moves by a large displacement. Thus, we design a strategy that only re-estimates the optical flow if the value of $flow(y)$ in (7) is less than a threshold. This speeds up the process significantly while maintaining good accuracy.

Second, since our goal is to find a stable object mask $\mathcal{M}$, instead of using the energy E_{seg} to decide the convergence status during the update of the segmentation model, we measure the difference of object mask solutions $\mathcal{M}$. If the overlap ratio of $\mathcal{M}$ is larger than a value (e.g. 95%), it should be a stable solution. In our experiments, the entire optimization process for (10) converges within five iterations, and converges in two iterations for most frames from our experiments.

5. Experimental Results

Implementation Details. To evaluate the proposed algorithm, we first construct the foreground and background color GMMs in the RGB space from the first frame, and set K to 5 for each GMM. For learning the online SVM model, we extract hierarchical CNN features [20] combining the first 5 convolutional layers from a pre-trained VGG net [28] into 1472 dimensional vectors. We use the method [13] to generate supervoxels and convert them to superpixels in each frame. For parameters in the graphical model, we use $\alpha_1^{col} = 1, \alpha_1^{cnn} = 3, \beta_1 = 2, \gamma_1^s = 3$ and $\gamma_1^t = 0.2$ on the pixel level. On the superpixel level, parameters are set as $\alpha_2 = 1, \beta_2 = 1$ and $\gamma_2 = 2$. For (4), we set $\lambda_1 = 1, \lambda_2 = 15$ and $\lambda_3 = 5$. Since one superpixel contains numerous pixels, we use larger weight for λ_2 on the superpixel level as otherwise the superpixel energy is easily absorbed to have the same label as the pixels (a similar issue holds for λ_3). All these parameters are fixed in the experiments. The MATLAB code will be made available at https://sites.google.com/site/yihsuantsai/.

SegTrack v2 Dataset. We first evaluate the proposed algorithm on the SegTrack v2 dataset [19] which consists of 14 videos with 24 objects and 947 annotated frames. The

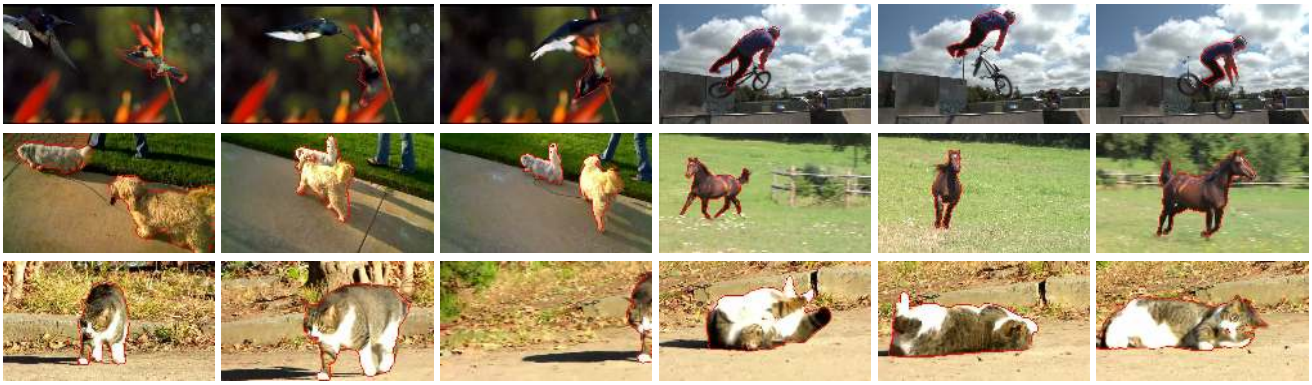


Figure 5. Example results for segmentation on the SegTrack v2 (first row) and Youtube-Objects (second and third rows) datasets. The output on the pixel level of our multi-level model is indicated as the red contour. The results show that our method is able to track and segment (multiple) objects under challenges such as occlusions, fast movements, deformed shapes and cluttered backgrounds. Best viewed in color with enlarged images.

Table 1. Segmentation results on the SegTrack v2 dataset with the overlap ratio. Note that "-" indicates that the method fails to track an object and is excluded in measuring accuracy.

Sequence/Object	[40]	[19]	[18]	[13]	[12]	[38]	Ours
Online?	√				√	√	√
Unsupervised?		√	√	√			
Girl	83.7	**89.2**	87.7	31.9	53.6	52.4	87.9
Birdfall	**77.5**	62.5	49.0	57.4	56.0	32.5	57.4
Parachute	94.4	93.4	**96.3**	69.1	85.6	69.9	94.5
Cheetah-Deer	**63.1**	37.3	44.5	18.8	46.1	33.1	33.8
Cheetah-Cheetah	35.3	40.9	11.7	24.4	47.4	14.0	**70.4**
Monkeydog-Monkey	**82.2**	71.3	74.3	68.3	61.0	22.1	54.4
Monkeydog-Dog	21.1	18.9	4.9	18.8	18.9	10.2	**53.3**
Penguin-#1	92.7	51.5	12.6	72.0	54.5	20.8	**93.9**
Penguin-#2	**91.8**	76.5	11.3	80.7	67.0	20.8	87.1
Penguin-#3	**91.9**	75.2	11.3	75.2	7.6	10.3	89.3
Penguin-#4	**90.3**	57.8	7.7	80.6	54.3	13.0	88.6
Penguin-#5	76.3	66.7	4.2	62.7	29.6	18.9	**80.9**
Penguin-#6	**88.7**	50.2	8.5	75.5	2.1	32.3	85.6
Drifting-#1	67.3	74.8	63.7	55.2	62.6	43.5	**84.3**
Drifting-#2	**63.7**	60.6	30.1	27.2	21.8	11.6	39.0
Hummingbird-#1	58.3	54.4	46.3	13.7	11.8	28.8	**69.0**
Hummingbird-#2	50.7	72.3	**74.0**	25.2	-	45.9	72.9
BMX-Person	**88.9**	85.4	87.4	39.2	2.0	27.9	88.0
BMX-Bike	5.7	24.9	**38.6**	32.5	-	6.0	7.0
Frog	61.9	72.3	0.0	67.1	14.5	45.2	**81.4**
Worm	76.5	82.8	84.4	34.7	36.8	27.4	**89.6**
Soldier	81.1	83.8	66.6	66.5	70.7	43.0	**86.4**
Monkey	86.0	84.8	79.0	61.9	73.1	61.7	**88.6**
Bird of Paradise	93.0	94.0	92.2	86.8	5.1	44.3	**95.2**
Mean per Object	71.8	65.9	45.3	51.8	40.1	30.7	**74.1**
Mean per Sequence	72.2	71.2	57.3	50.8	41.0	37.0	**75.3**

dataset includes different challenging sequences with large appearance change, occlusion, motion blur, complex deformation and interaction between objects. For videos containing multiple objects, since instance-level annotations are provided, each object can be segmented in turn, treating each as a problem of segmenting that object from the background. We first present segmentation results and demonstrate the effectiveness of the multi-level model.

Table 1 shows segmentation accuracy of the proposed algorithm and the state-of-the-art methods including tracking and graph based approaches [12, 13, 18, 19, 38, 40]. Note that our model can generate labeled results on both pixel and superpixel levels, and we present the pixel level fine-grained segmentation results. We use the intersection-over-union (overlap) ratio for evaluation as it has been shown that the pixel error metric used in the SegTrack dataset is sensitive to object size [19] and is less informative.

Overall, the proposed algorithm achieves favorable results in most sequences especially for non-rigid objects (*Hummingbird, Worm, Soldier*). These sequences usually contain large deformation due to fast motions or complex cluttered backgrounds. The superpixel-based tracking methods [38, 40] do not perform well on these sequences since superpixels may not preserve object boundaries well. The Hough-based tracking method [12] only uses pixels, which may result in noisy temporal links. In the proposed spatial-temporal multi-level model, we consider both pixels and superpixels in the tracking and segmentation to maintain object boundaries as well as temporal consistency.

For the *Penguin* and *Frog* sequences, the object appearance is similar to the background with slow motions. The off-line methods [18, 19] that generate object proposals from all frames, are likely to have large segmentation errors due to wrong association or incomplete regions that contain foreground and background pixels. In contrast, the proposed algorithm performs well in these sequences with objects surrounded by other objects or background with similar appearance. In the location term (7), our model considers consistency between supervoxels and optical flow, and this helps maintain temporal consistency. We present qualitative segmentation results in Figure 5 and show more results in the supplementary material.

Table 2. Segmentation results using multi-level and single-level models on the SegTrack v2 dataset with the overlap ratio. Note that there are results on both pixel and superpixel levels using the multi-level model.

Methods	Pixel multi-level	Pixel only	Superpixel multi-level	Superpixel only
Mean per Object	**74.1**	69.6	65.6	50.3

Table 3. Segmentation results on the Youtube-Objects dataset with the overlap ratio.

Category	[22]	[14]	[36]	[12]	[24]	[23]	Ours
aeroplane	89.0	86.3	79.9	73.6	70.9	13.7	**89.9**
bird	81.6	81.0	78.4	56.1	70.6	12.2	**84.2**
boat	**74.2**	68.6	60.1	57.8	42.5	10.8	74.0
car	70.9	69.4	64.4	33.9	65.2	23.7	**80.9**
cat	67.7	58.9	50.4	30.5	52.1	18.6	**68.3**
cow	79.1	68.6	65.7	41.8	44.5	16.3	**79.8**
dog	70.3	61.8	54.2	36.8	65.3	18.0	**76.6**
horse	67.8	54.0	50.8	44.3	53.5	11.5	**72.6**
motorbike	61.5	60.9	58.3	48.9	44.2	10.6	**73.7**
train	**78.2**	66.3	62.4	39.2	29.6	19.6	76.3
Mean	74.0	67.6	62.5	46.3	53.8	15.5	**77.6**

To evaluate the proposed multi-level model that integrates both levels, we compare to our model using only pixels or superpixels in Table 2. The information contained on these two levels complement each other as the one based on the superpixel level maintains local region consistency, while the one based on the pixel level refines incomplete object contours (See Figure 4). Specifically, the location term in (7) enhances temporal information such that the model can handle cases including fast movements and background noise in sequences. In addition, the proposed multi-level model with superpixels performs better than that using only superpixels, which demonstrates that the refinement with the pixel level information is critical for obtaining good performance especially in sequences that contain unclear object boundaries. We also note that our single-level models already perform comparably to the state-of-the-art methods.

Youtube-Objects Dataset. The Youtube-Objects dataset [25] contains 1407 videos with 10 object categories, and the length of the sequences is up to 400 frames. We evaluate the proposed algorithm in a subset of 126 videos with more than 20000 frames, where the pixel-wise annotations in every 10 frames are provided by Jain and Grauman [14]. Table 3 shows the segmentation results of the proposed algorithm and other state-of-the-art methods[1]. For tracking-based or foreground propagation algorithms [12, 14, 22, 23, 36], ground truth annotations in the first frame are used as initializations to propagate segmentation masks. For multiple objects in videos, the proposed algorithm is able to propagate multiple object segmentations at the same time. Note that there are no instance-level annotations provided.

Overall, the proposed algorithm performs well in terms of overlap ratio, especially in 8 out of 10 categories. Compared to optical flow based methods [22, 36], the proposed algorithm performs well on fast moving objects such as *car* and *motorbike* as the optical flow errors are reduced. Although the recent method [14] utilizes long-term supervoxels to enhance the temporal connection, the segmentation results contain noisy object boundaries as only superpixels are used. In contrast, the proposed algorithm considers visual information at multiple levels and delineates boundaries well especially on non-rigid objects (*dog, horse, cow*). We show qualitative results in Figure 5. More results are presented in the supplementary material.

Optical Flow. We demonstrate the effectiveness of iteratively optimizing two models for updated optical flow and segmentation results (See Algorithm 1) on the SegTrack v2 dataset. Here, we only consider sequences with large displacements in which the flow is re-estimated. First, we measure the quality of updated optical flow. As the optical flow ground truth is not available, we warp the object segmentation ground truth from frame t to frame $t-1$ using optical flow with the bicubic interpolation. We then compute the overlap ratio between the warped and ground truth masks. Since we focus on optical flow of the object, this metric measures consistency for flow directions and whether they connect to the same objects between two frames.

Table 4 shows results compared to two other optical flow methods [6, 30] and the layered model [32]. The updated results improve the optical flow estimation [30] consistently in all the sequences, especially for fast moving objects (*Girl, Drifting, BMX*). This validates our approach since the method [30] is used to compute the initial flow. Figure 6 illustrates the optical flow results. Compared to the other three methods, the updated optical flow exhibits clearer object boundaries. It shows the importance of computing optical flow on local object regions. In contrast, the results from [6] are usually oversmoothed around the object boundaries, and the layered model [32] generates incomplete flows inside objects.

In addition, we use the normalized interpolation error (NE) as described in [2] for evaluation. Similarly, the ground truth of the colored object is warped by the optical flow from frame t to $t-1$. The average NE of the updated optical flow is better than [30], but slightly worse than [6]. It can be attributed to the fact that the oversmoothed optical flow in [6] usually generates more complete warped images after interpolation; this is favored by the NE metric.

Second, by using the updated optical flow, we re-estimate object segmentations and measure overlap ratios. The updated segmentation results are improved for all the

[1]We evaluate the code of [40] released by the authors. However, the algorithm requires different parameter settings for challenging sequences. We discuss and report results in the supplementary material.

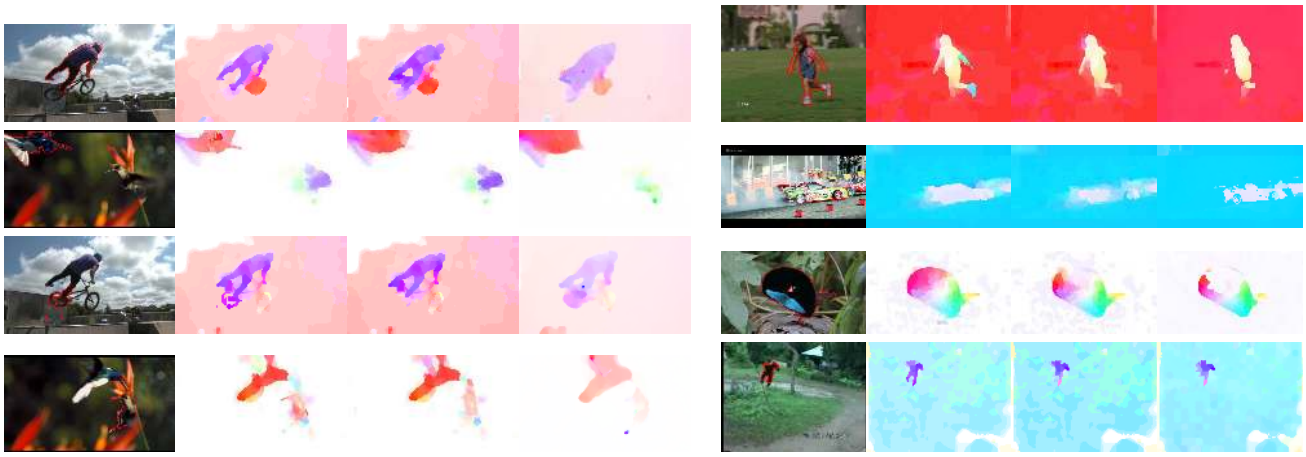


Figure 6. Results for updated optical flow on the SegTrack v2 dataset. We present our updated optical flow compared to the initial flow [30], and the other two methods, Brox [6] on the left and Sun [32] on the right. Our results contain object boundaries guided by the segmented object marked with the red contour. Note that in the same sequence with multiple objects, the updated optical flow varies depending on the segmentation. Best viewed in color with enlarged images.

Table 4. Intersection-over-union ratio for warped images by interpolation and updated optical flow on the SegTrack v2 dataset. The last row shows the average of normalized interpolation error. The performance is evaluated on frames with sigificant motion.

Sequence/Object	Ours	Sun [30]	Brox [6]	Sun [32]
Girl	**64.6**	56.1	63.2	**64.6**
Parachute	**86.8**	84.9	83.2	83.3
MonkeyDog-Monkey	**70.8**	70.0	67.4	69.0
MonkeyDog-Dog	69.0	69.0	68.9	**75.0**
Drifting-#1	89.5	86.6	**91.3**	82.5
Drifting-#2	87.3	82.5	**87.7**	79.8
Hummingbird-#1	**55.2**	52.2	52.6	51.3
Hummingbird-#2	**71.1**	70.6	68.7	65.5
Worm	73.3	71.1	69.8	**91.1**
Monkey	77.4	76.3	**80.9**	70.5
Soldier	**84.5**	83.5	80.8	82.7
Bird of Paradise	**94.4**	87.9	88.3	89.7
BMX-Person	**80.0**	77.4	75.0	72.2
BMX-Bike	**38.4**	33.9	37.5	38.3
Mean	**74.5**	71.6	72.5	72.5
Average NE	0.36	0.38	**0.32**	0.37

sequences, and the average overlap ratio is increased from 72.9% to 75.1% in sequences that rely on the optical flow. The improvement varies in different sequences since the segmentation model also takes other cues such as appearance and location into account. For instance, the improvement of overlap ratio is limited in the *Bird of Paradise* and *Parachute* sequences since the objects move steadily. On the other hand, for objects with noisy cluttered backgrounds (*Drifting-#2*) or with similar appearance to the background regions (*Worm*), the overlap ratios are improved by 2.4% and 2.9% respectively. More results and comparisons are provided in the supplementary material.

Runtime Performance. Our MATLAB implementation of object flow takes 3 to 20 seconds per frame on the SegTrack v2 dataset depending on the object size, and on average it takes 12.2 seconds per frame. In contrast, the state-of-the-art method [40] takes 59.6 seconds per frame on average. Note that all the timings are measured on the same computer with 3.60 GHz Intel i7 CPU and 32 GB memory, and exclude the time to compute optical flow as each method uses different algorithms. We use the MATLAB implementation of [30] to generate optical flow fields (around 30 seconds per frame) and it could be replaced by faster algorithms [33, 41].

6. Concluding Remarks

In this paper, we present a novel algorithm for joint optimization of segmentation and optical flow in videos, and show that the problem can be efficiently solved. For segmentation, a multi-level model containing pixels and superpixels is utilized to track objects. We show that both levels complement each other and maintain object boundaries and temporal consistency throughout the video. Using the segmentation, we modify the optical flow estimation to be performed within local foreground and background regions, resulting in more accurate optical flow, particularly around object boundaries. We show that our method performs favorably against state-of-the-art methods on two datasets, and both the segmentation and optical flow results are improved by iteratively updating both models.

Acknowledgments. This work is supported in part by the NSF CAREER Grant #1149783 and NSF IIS Grant #1152576, and portions of this work were performed while Y.-H. Tsai was an intern at MPI.

References

[1] V. Badrinarayanan, F. Galasso, and R. Cipolla. Label propagation in video sequences. In *CVPR*, 2010. 1, 2

[2] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. In *ICCV*, 2007. 1, 7

[3] M. J. Black and A. Jepson. Estimating optical flow in segmented images using variable-order parametric models with local deformations. *PAMI*, 18(10):972–986, Oct. 1996. 1

[4] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, pages 1124–1137, 2004. 5

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

[6] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *PAMI*, 33(3):500–13, 2011. 1, 7, 8

[7] J. Chang and J. W. Fisher. Topology-constrained layered tracking with latent flow. In *ICCV*, 2013. 2

[8] J. Chang, D. Wei, and J. W. Fisher. A video representation using temporal superpixels. In *CVPR*, 2013. 1

[9] Z. Chen, H. Jin, Z. Lin, S. Cohen, and Y. Wu. Large displacement optical flow from nearest neighbor fields. In *CVPR*, 2013. 1

[10] P. Chockalingam, N. Pradeep, and S. Birchfield. Adaptive fragments-based tracking of non-rigid objects using level sets. In *ICCV*, 2009. 1

[11] F. Galasso, N. Nagaraja, T. Cardenas, T. Brox, and B. Schiele. A unified video segmentation benchmark: Annotation, metrics and analysis. In *ICCV*, 2013. 1

[12] M. Godec, P. M. Roth, and H. Bischof. Hough-based tracking of non-rigid objects. In *ICCV*, 2011. 1, 2, 6, 7

[13] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *CVPR*, 2010. 1, 2, 4, 5, 6

[14] S. D. Jain and K. Grauman. Supervoxel-consistent foreground propagation in video. In *ECCV*, 2014. 1, 2, 7

[15] A. Jepson and M. J. Black. Mixture models for optical flow computation. In *CVPR*, 1993. 2

[16] N. Jojic and B. Frey. Learning flexible sprites in video layers. In *CVPR*, 2001. 2

[17] C. Lalos, H. Grabner, L. V. Gool, and T. Varvarigou. Object flow: Learning object displacement. In *ACCV workshop*, 2010. 2

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

[19] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, 2013. 1, 2, 5, 6

[20] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang. Hierarchical convolutional features for visual tracking. In *ICCV*, 2015. 3, 5

[21] T. Ma and L. J. Latecki. Maximum weight cliques with mutex constraints for video object segmentation. In *CVPR*, 2012. 1

[22] N. S. Nagaraja, F. Schmidt, and T. Brox. Video segmentation with just a few strokes. In *ICCV*, 2015. 7

[23] P. Ochs, J. Malik, and T. Brox. Segmentation of moving objects by long term video analysis. *PAMI*, 36(6):1187–1200, 2014. 1, 7

[24] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *ICCV*, 2013. 1, 2, 3, 7

[25] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, 2012. 2, 7

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

[27] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, 2004. 2

[28] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556:1187–1200, 2014. 5

[29] D. Sun, C. Liu, and H. Pfister. Local layering for joint motion estimation and occlusion detection. In *CVPR*, 2013. 1

[30] D. Sun, S. Roth, and M. J. Black. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *IJCV*, 106(2):115–137, 2014. 1, 5, 7, 8

[31] D. Sun, E. B. Sudderth, and M. J. Black. Layered segmentation and optical flow estimation over time. In *CVPR*, 2012. 2

[32] D. Sun, J. Wulff, E. B. Sudderth, H. Pfister, and M. J. Black. A fully-connected layered model of foreground and background flow. In *CVPR*, 2013. 2, 7, 8

[33] N. Sundaram, T. Brox, and K. Keutzer. Dense point trajectories by gpu-accelerated large displacement optical flow. In *ECCV*, 2010. 8

[34] P. H. S. Torr, R. Szeliski, and P. Anandan. An integrated Bayesian approach to layer extraction from image sequences. *PAMI*, 23(3):297–303, Mar. 2001. 2

[35] D. Tsai, M. Flagg, and J. M. Rehg. Motion coherent tracking with multi-label mrf optimization. In *BMVC*, 2010. 1, 2

[36] S. Vijayanarasimhan and K. Grauman. Active frame selection for label propagation in videos. In *ECCV*, 2012. 1, 2, 7

[37] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *TIP*, 3:625–638, 1994. 2

[38] S. Wang, H. Lu, F. Yang, and M.-H. Yang. Superpixel tracking. In *ICCV*, 2011. 1, 2, 6

[39] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. Deepflow: Large displacement optical flow with deep matching. In *ICCV*, 2013. 1

[40] L. Wen, D. Du, Z. Lei, S. Z. Li, and M.-H. Yang. Jots: Joint online tracking and segmentation. In *CVPR*, 2015. 1, 2, 6, 7, 8

[41] J. Wulff and M. J. Black. Efficient sparse-to-dense optical flow estimation using a learned basis and layers. In *CVPR*, 2015. 2, 8

[42] C. Xu, C. Xiong, and J. J. Corso. Streaming hierarchical video segmentation. In *ECCV*, 2012. 1, 2

[43] L. Xu, J. Chen, and J. Jia. A segmentation based variational model for accurate optical flow estimation. In *ECCV*, 2008. 1

[44] L. Xu, J. Jia, and Y. Matsushita. Motion detail preserving optical flow estimation. In *CVPR*, 2010. 1

[45] D. Zhang, O. Javed, and M. Shah. Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In *CVPR*, 2013. 1

[46] Y. Zhou and H. Tao. A background layer model for object tracking through occlusion. In *CVPR*, 2003. 2

[47] C. L. Zitnick, N. Jojic, and S. B. Kang. Consistent segmentation for optical flow estimation. In *ICCV*, 2005. 1