Unsupervised Video Object Segmentation using Motion Saliency-Guided Spatio-Temporal Propagation

Yuan-Ting Hu[1], Jia-Bin Huang[2], and Alexander G. Schwing[1]

[1]University of Illinois at Urbana-Champaign [2]Virginia Tech
{ythu2,aschwing}@illinois.edu jbhuang@vt.edu

Abstract. Unsupervised video segmentation plays an important role in a wide variety of applications from object identification to compression. However, to date, fast motion, motion blur and occlusions pose significant challenges. To address these challenges for unsupervised video segmentation, we develop a novel saliency estimation technique as well as a novel neighborhood graph, based on optical flow and edge cues. Our approach leads to significantly better initial foreground-background estimates and their robust as well as accurate diffusion across time. We evaluate our proposed algorithm on the challenging DAVIS, SegTrack v2 and FBMS-59 datasets. Despite the usage of only a standard edge detector trained on 200 images, our method achieves state-of-the-art results outperforming deep learning based methods in the unsupervised setting. We even demonstrate competitive results comparable to deep learning based methods in the semi-supervised setting on the DAVIS dataset.

1 Introduction

Unsupervised foreground-background video object segmentation of complex scenes is a challenging problem which has many applications in areas such as object identification, security, and video compression. It is therefore not surprising that many efforts have been devoted to developing efficient techniques that are able to effectively separate foreground from background, even in complex videos.

In complex videos, cluttered backgrounds, deforming shapes, and fast motion are major challenges. In addition, in the unsupervised setting, algorithms have to automatically discover foreground regions in the video. To this end, classical video object segmentation techniques [6, 9, 11, 18, 23, 46, 22, 50, 58] often assume rigid background motion models and incorporate a scene prior, two assumptions which are restrictive in practice. Trajectory based methods, such as [8, 12, 45, 5, 15], require selection of clusters or a matrix rank, which may not be intuitive. Graphical model based approaches [24, 2, 16, 51, 54, 52] estimate the foreground regions using a probabilistic formulation. However, for computational efficiency, the constructed graph usually contains only local connections, both spatially and temporally, reducing the ability to consider long-term spatial and temporal coherence patterns. To address this concern, diffusion based methods [35], *e.g.*, [13, 55], propagate an initial foreground-background estimate more globally. While promising results are shown, diffusion based formulations rely

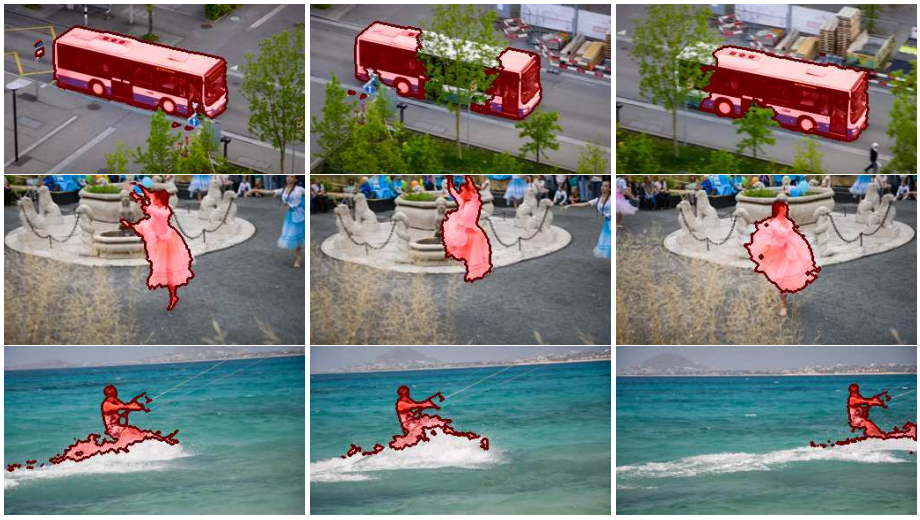

Fig. 1: **Video object segmentation in challenging scenarios.** Given an input video, our algorithm produces accurate segmentation of the foreground object *without any manual annotations*. Our method is capable of handling unconstrained videos that span a wide variety of situations including occlusion (BUS), non-ridge deformation (DANCE-JUMP), and dynamic background (KITE-SURF).

heavily on the initialization as well as an accurate neighborhood graph encoding the semantic distance between pixels or superpixels.

Therefore, in this paper, we develop (1) a new initialization technique and (2) a more robust neighborhood graph. Our initialization technique is based on the intuition that the optical flow on the boundary of an image differs significantly from the moving direction of the object of interest. Our robust neighborhood graph is built upon accurate edge detection and flow cues.

We highlight the performance of our proposed approach in Figure 1 using three challenging video sequences. Note the fine details that our approach is able to segment despite the fact that our method is unsupervised. Due to accurate initial estimates and a more consistent neighborhood graph, we found our method to be robust to different parameter choices. Quantitatively, our initialization technique and neighborhood graph result in significant improvements for unsupervised foreground-background video segmentation when compared to the current state-of-the-art. On the recently released DAVIS dataset [42], our unsupervised non-deep learning based segmentation technique outperforms current state-of-the-art methods by more than 1.3% in the unsupervised setting. Our method also achieves competitive performance compared with deep net based techniques in the semi-supervised setting.

2 Related Work

The past decade has seen the rapid development in video object segmentation [51, 33, 38, 44, 32, 31, 40, 57, 17, 52, 25, 19, 20]. Given different degrees of human interaction, these methods model inter- and intra-frame relationship of the pixels or superpixels to determine the foreground-background labeling of the observed scene. Subsequently, we classify the literature into four areas based on the degree of human involvement and discuss the relationship between video object and video motion segmentation.

Unsupervised video object segmentation: Fully automatic approaches for video object segmentation have been explored recently [7, 31, 59, 40, 39, 13, 57, 30], and no manual annotation is required in this setting. Unsupervised foreground segmentation discovery can be achieved by motion analysis [40, 13], trajectory clustering [39], or object proposal ranking [31, 57]. Our approach computes motion saliency in a given video based on boundary similarity of motion cues. In contrast, Faktor and Irani [13] find motion salient regions by extracting dominant motion. Subsequently they obtain the saliency scores by computing the motion difference with respect to the detected dominant motion. Papazoglou and Ferrari [40] identify salient regions by finding the motion boundary based on optical flow and computing inside-outside maps to detect the object of interest.

Recently, deep learning based methods [25, 49, 48] were also used to address unsupervised video segmentation. Although these methods do not require the ground truth of the first frame of the video (unsupervised as opposed to semi-supervised), they need a sufficient amount of labeled data to train the models. In contrast, our approach works effectively in the unsupervised setting and does not require training data beyond the one used to obtain an accurate edge detector.

Tracking-based video object segmentation: In this setting, the user annotation is reduced to only one mask for the first frame of the video [4, 17, 51, 24, 52, 36, 41]. These approaches track the foreground object and propagate the segmentation results to successive frames by incorporating cues such as motion [51, 52] and supervoxel consistency [24]. Again, our approach differs in that we don't consider any human labels.

Interactive video object segmentation: Interactive video object segmentation allows users to annotate the foreground segments in key frames to generate impressive results by propagating the user-specified masks across the entire video [44, 14, 38, 34, 24]. Price *et al*. [44] further combine multiple features, of which the weights are automatically selected and learned from user inputs. Fan *et al*. [14] tackle interactive segmentation by enabling bi-directional propagation of the masks between non-successive frames. Our approach differs in that the proposed method does not require any human interaction.

Video motion segmentation: Video motion segmentation [5] aims to segment a video based on motion cues, while video object segmentation aims at segmenting the foreground based on objects. The objective function differs: for motion segmentation, clustering based methods [5, 39, 29, 32] are predominant and group point trajectories. In contrast, for video object segmentation, a binary labeling formulation is typically applied as we show next by describing our approach.

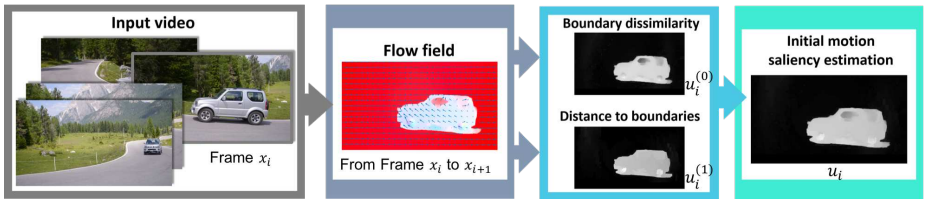

Fig. 2: **Motion saliency estimation**. Given an input video, we compute the flow field for each frame. We detect the saliency score based on the flow vector by calculating a boundary dissimilarity map $u^{(0)}$ and a distance map $u^{(1)}$ indicating the distance of each pixel to the boundaries. We use minimum barrier distance to measure the distance. The motion saliency estimation is computed by averaging the boundary dissimilarity map and the distance map.

3 Unsupervised Video Object Segmentation

The two most important ingredients for unsupervised video object segmentation are the initial saliency estimate as well as a good assessment of the neighborhood relation of pixels or superpixels. For initial saliency prediction in unsupervised video object segmentation we describe a novel method comparing the motion at a pixel to the boundary motion. Intuitively, boundary pixels largely correspond to background and pixels with a similar motion are likely background too. To construct a meaningful neighborhood relation between pixels we assess flow and appearance cues. We provide details for both contributions after describing an overview of our unsupervised video object segmentation approach.

Method overview: Our method uses a diffusion mechanism for unsupervised video segmentation. Hence, the approach distributes an initial foreground saliency estimate over the F frames x_i, $i \in \{1,\dots,F\}$, of a video $x = (x_1,\dots,x_F)$. To this end, we partition each frame into a set of nodes using superpixels, and estimate and encode their semantic relationship within and across frames using a global neighborhood graph. Specifically, we represent the global neighborhood graph by a weighted row-stochastic adjacency matrix $G \in \mathbb{R}^{N\times N}$, where N is the total number of nodes in the video. Diffusion of the initial foreground saliency estimates $v^0 \in \mathbb{R}^N$ for each node is performed by repeated matrix multiplication of the current node estimate with the adjacency matrix G, *i.e.*, for the t-th diffusion step $v^t = Gv^{t-1}$.

With the adjacency matrix G and initialization v^0 being the only inputs to the algorithm, it is obvious that they are of crucial importance for diffusion based unsupervised video segmentation. We focus on both points in the following and develop first a new saliency estimation of v^0 before discussing construction of the neighborhood graph G.

3.1 Saliency estimation

For unsupervised video object segmentation, we propose to estimate the motion saliency by leveraging a boundary condition. Since we are dealing with video, motion is one of the most important cues for identifying moving foreground objects. In general, the

motion of the foreground object differs from background motion. But importantly, the background region is often connected to the boundary of the image. While the latter assumption is commonly employed for *image saliency* detection, it has not been exploited for *motion saliency* estimation. To obtain the initial saliency estimate v^0 defined over superpixels, we average the pixelwise motion saliency results u over the spatial support of each superpixel. We subsequently describe our developed procedure for foreground saliency estimation, taking advantage of the boundary condition. The proposed motion saliency detection is summarized in Figure 2.

Conventional motion saliency estimation techniques for video object segmentation are based on either background subtraction [6], trajectory clustering [5], or motion separation [13]. Background subtraction techniques typically assume a static camera, which is not applicable for complex videos. Trajectory clustering groups points with similar trajectories, which is sensitive to non-rigid transformation. Motion separation detects background by finding the dominant motion and subsequently calculates the difference in magnitude and/or orientation between the motion at each pixel, and the dominant motion. The larger the difference, the more likely the pixel to be foreground. Again, complex motion poses challenges, making it hard to separate foreground from background.

In contrast, we propose to use the boundary condition that is commonly used for *image saliency* detection [56, 53] to support *motion saliency* estimation for unsupervised video segmentation. Our approach is based on the intuition that the background region is connected to image boundaries in some way. Therefore we calculate a distance metric for every pixel to the boundary. Compared to the aforementioned techniques, we will show that our method can better deal with complex, non-rigid motion.

We use u to denote the foreground motion saliency of the video. Moreover, u_i and $u_i(p_i)$ denote the foreground saliency for frame i and for pixel p_i in frame i respectively. To compute the motion saliency estimate, we treat every frame x_i, $i \in \{1,\dots,F\}$ independently. Given a frame x_i, let $x_i(p_i)$ refer to the intensity values of pixel p_i, and let $f_i(p_i) \in \mathbb{R}^2$ denote the optical flow vector measuring the motion of the object illustrated at pixel p_i between frame i and frame $i+1$. In addition, let $\mathscr{B}_i$ denote the set of boundary pixels of frame i.

We compute the foreground motion saliency u_i of frame i based on two terms $u_i^{(0)}$ and $u_i^{(1)}$, each of which measures a distance between any pixel p_i of the i-th frame and the boundary $\mathscr{B}_i$. For the first distance $u_i^{(0)}$, we compute the smallest flow direction difference observed between a pixel p_i and common flow directions on the boundary. For the second distance $u_i^{(1)}$, we measure the smallest barrier distance between pixel p_i and boundary pixels. Both of the terms capture the similarity between the motion at pixel p_i and the background motion. Subsequently, we explain both terms in greater detail.

Computing flow direction difference: More formally, to compute $u_i^{(0)}(p_i)$, the flow direction difference between pixel p_i in frame i and common flow directions on the boundary $\mathscr{B}_i$ of frame i, we first cluster the boundary flow directions into a set of K

clusters $k \in \{1,\dots,K\}$ using k-means. We subsume the cluster centers in the set

$$\mathscr{K}_i = \left\{ \mu_{i,k} : \mu_{i,k} = \arg\min_{\hat{\mu}_{i,k}} \min_{r \in \{0,1\}^{|\mathscr{B}_i|K}} \frac{1}{2} \sum_{p_i \in \mathscr{B}_i,k} r_{p_i,k} \| f_i(p_i) - \hat{\mu}_{i,k} \|_2^2 \right\}. \tag{1}$$

Hereby, $r_{p_i,k} \in \{0,1\}$ is an indicator variable which assigns pixel p_i to cluster k, and r is the concatenation of all those indicator variables. We update $\mathscr{K}_i$ to only contain centers with more than $1/6$ of the boundary pixels assigned. Given those cluster centers, we then obtain a first distance measure capturing the difference of flow between pixel p_i in frame i and the major flow directions observed at the boundary of frame i via

$$u_i^{(0)}(p_i) = \min_{\mu_{i,k} \in \mathscr{K}_i} \| f_i(p_i) - \mu_{i,k} \|_2^2. \tag{2}$$

Computing smallest barrier distance: When computing the smallest barrier distance $D_{bd,i}$ between pixel p_i in frame i and boundary pixels, *i.e.*, to obtain

$$u_i^{(1)}(p_i) = \min_{s \in \mathscr{B}_i} D_{bd,i}(p_i, s), \tag{3}$$

we use the following barrier distance:

$$D_{bd,i}(p_i, s) = \max_{e \in \Pi_{i,p_i,s}} w_i(e) - \min_{e \in \Pi_{i,p_i,s}} w_i(e). \tag{4}$$

Hereby, $\Pi_{i,p_i,s}$ denotes the path, *i.e.*, a set of edges connecting pixel p_i to boundary pixel $s \in \mathscr{B}_i$, obtained by computing a minimum spanning tree on frame i. The edge weights $w_i(e)$, which are used to compute both the minimum spanning tree as well as the barrier distance given in Eq. (4), are obtained as the maximum flow direction difference between two neighboring pixels, *i.e.*, $w_i(e) = \max\{f_i(p_i) - f_i(q_i)\} \in \mathbb{R}$ where the max is taken across the two components of $f_i(p_i) - f_i(q_i) \in \mathbb{R}^2$. Note that $e = (p_i, q_i)$ refers to an edge connecting the two pixels p_i and q_i. To compute the minimum spanning tree we use the classical 4-connected neighborhood. Intuitively, we compute the barrier distance between 2 points as the difference between the maximum edge weight and minimum edge weight on the path of the minimum spanning tree between the 2 points. We then compute the smallest barrier distance of a point as the minimum of the barrier distances between the point and any point on the boundary.

Computing foreground motion saliency: We obtain the pixelwise foreground motion saliency u_i of frame i when adding the two distance metrics $u_i^{(0)}$ and $u_i^{(1)}$ after having normalized each of them to a range of $[0,1]$ by subtracting the minimum entry in $u_i^{(\cdot)}$ and dividing by the difference between the maximum and minimum entry. Examples for $u_i^{(0)}$, $u_i^{(1)}$ and the combined motion saliency are visualized in Figure 2.

We found the proposed changes to result in significant improvements for saliency estimation of video data. We present a careful assessment in Section 4.

3.2 Neighborhood construction

The second important term for diffusion based video segmentation beyond initial estimates is the neighborhood graph G. Classical techniques construct the adjacency matrix

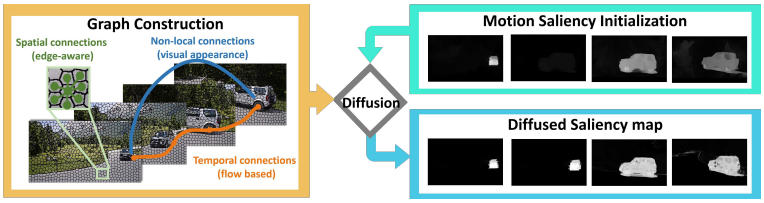

Fig. 3: **Graph construction**. In our method, we construct a graph for diffusing the initial motion saliency estimation. Our graph contains 1) edge-aware spatial connections (intra-frame connections), 2) flow-based temporal connections (inter-frame connections and 3) non-local long range connections. We show the initial motion saliency and the diffused saliency map using the constructed graph. We found these three types of connections to help propagate the initial saliency estimation effectively.

using local information, such as connecting a node with its spatial and temporal neighbors, and non-local connections. These methods establish a connection between two nodes as long as their visual appearance is similar.

In contrast, we compute the neighborhood graph, *i.e.*, the adjacency matrix for graph diffusion, $G = T \times E \times V$ as the product of three components, based on inter-frame information T, intra-frame signals E, and long-range components V, as shown in Figure 3, and use a variety of cues for robustness. We formally discuss each of the components in the following.

Inter-frame temporal information is extracted from optical flow cues. We connect superpixels between adjacent frames following flow vectors while checking the forward/backward consistency in order to prevent inaccurate flow estimation at motion boundaries.

More formally, to compute the flow adjacency matrix T, consider two successive video frames x_i and x_{i+1} each containing pixel p_i and p_{i+1}, respectively. We compute a forward flow field $f_i(p_i)$ and a backward flow field $b_{i+1}(p_{i+1})$ densely for every pixel p using [21]. Using those flow fields, we define the forward confidence score $c_i^F(p_i)$ at pixel p_i of frame x_i via

$$c_i^F(p_i) = \exp\left(\frac{-\| - f_i(p_i) - b_{i+1}(p_i + f_i(p_i))\|_2^2}{\sigma_2}\right), \tag{5}$$

and the backward confidence score $c_i^B(p_i)$ at pixel p_i of frame x_i via

$$c_i^B(p_i) = \exp\left(\frac{-\| - b_i(p_i) - f_{i-1}(p_i + b_i(p_i))\|_2^2}{\sigma_2}\right), \tag{6}$$

where σ_2 is a hyper-parameter. Intuitively, this confidence score measures the distance between the pixel p_i and the result obtained after following the flow field into frame x_{i+1} via $p_i + f_i(p_i)$ and back into frame x_i via $p_i + f_i(p_i) + b_{i+1}(p_i + f_i(p_i))$. Taking the difference between pixel p_i and the obtained reprojection results in the term given in Eq. (5) and Eq. (6). We use the confidence scores to compute the connection strength

between two superpixels $s_{i,k}$ and $s_{i+1,m}$ in frame i and $i+1$ via

$$T(s_{i,k}, s_{i+1,m}) = \sum_{p \in s_{i,k}} \frac{\delta(p + f_i(p) \in s_{i+1,m}) c_i^F(p)}{|s_{i,k}| + |s_{i+1,m}|} + \sum_{p' \in s_{i+1,m}} \frac{\delta(p' + b_{i+1}(p') \in s_{i,k}) c_{i+1}^B(p')}{|s_{i,k}| + |s_{i+1,m}|}. \quad (7)$$

Hereby $\delta(\cdot)$ denotes the indicator function and $|s_{i,k}|$ and $|s_{i+1,m}|$ represent the number of pixels in $s_{i,k}$ and $s_{i+1,m}$, respectively. Intuitively, the first term compares the strength of the connections that start in superpixel $s_{i,k}$ and end up in superpixel $s_{i+1,m}$ with the total amount of strength originating from both $s_{i,k}$ and $s_{i+1,m}$. Similarly for the second term.

Intra-frame spatial information prevents diffusion across visual edges within a frame, while allowing information to be propagated between adjacent superpixels in the same frame if they aren't separated by a strong edge.

More formally, to find the edge aware spatial connections E, we first detect the edge responses frame-by-frame using the training based method discussed in [10]. Given edge responses, we calculate the confidence scores $A(s)$ for all superpixel s by summing over the decay function, *i.e.*,

$$A(s) = \frac{1}{|s|} \sum_{p \in s} \frac{1}{1 + exp(\sigma_w \cdot (G(p) - \varepsilon))}. \quad (8)$$

Hereby, $G(p) \in [0,1]$ is the edge response at pixel p. σ_w and ε are hyper-parameters, which we fix at $\sigma_w = 50$ and $\varepsilon = 0.05$ for all our experiments.

We calculate the edge-aware adjacency matrix E by exploiting the above edge information. Specifically,

$$E(s_{i,k}, s_{i,m}) = \frac{1}{2} \left(A(s_{i,k}) + A(s_{i,m}) \right), \quad (9)$$

if $s_{i,k}$ is spatially close to $s_{i,m}$, *i.e.*, if the distance between the centers of the two superpixels is less than 1.5 times the square root of the size of the superpixel.

Long range connections based on visual similarity allow propagating information between superpixels that are far away either temporally or spatially as long as the two are visually similar. These long-range connections enable the information to propagate more efficiently through the neighborhood graph.

More formally, to compute the visual similarity matrix V, we find those superpixels that are most closely related to a superpixel $s_{i,m}$. To this end, we first perform a k nearest neighbor search. More specifically, for each superpixel $s_{i,m}$ we find its k nearest neighbors that are within a range of r frames temporally. To compute the distance between two superpixels we use the Euclidean distance in the feature space.

We compute features $f(s)$ of a superpixel s by concatenating the LAB and RGB histograms computed over the pixels within a superpixel. We also include the HOG feature, and the x and y coordinate of the center of the superpixel.

Let the k nearest neighbors of the superpixel $s_{i,m}$ be referred to via $N(s_{i,m})$. The visual similarity matrix is then defined via

$$V(s_{i,m}, s) = \exp\left(\frac{-\|f(s_{i,m}) - f(s)\|_2^2}{\sigma} \right) \quad \forall s \in N(s_{i,m}), \quad (10)$$

where σ is a hyper-parameter and $f(s)$ denotes the feature representation of the superpixel s. Note that we use the same features to find k nearest neighbors and to compute the visual similarity matrix V. In this work, we refrain from using deep net based information even though we could easily augment our technique with more features.

To address the computational complexity, we use an approximate k nearest neighbor search. Specifically, we use the fast implementation of ANN search utilizing the randomized k-d forest provided in [37].

4 Experiments

In the following, we present the implementation details, describe the datasets and metrics used for evaluation, followed by ablation study highlighting the influences of the proposed design choices and comparisons with the state-of-the-art.

4.1 Implementation details

For the proposed saliency estimation algorithm, we set the number of clusters $K = 3$ for modeling the background. For neighborhood graph construction described in Section 3.2, we found $k = 40, r = 15, \sigma = 0.1, \sigma_2 = 2^{-6}, \sigma_w = 50$ to work well across datasets. The number of diffusion iterations is set to 25. In the supplementary material, we show that the performance of our method is reasonably robust to parameter choices.

The average running time of our approach on the DAVIS dataset, including the graph construction and diffusion is about 8.5 seconds per frame when using a single PC with Intel i7-4770 CPU and 32 GB memory. Extracting superpixels and feature descriptors takes about 1.5 and 0.8 seconds per frame, respectively. We use the implementation by [21, 47] for computing optical flow, which takes about 10.7 seconds per frame, including both forward flow and backward flow.

4.2 Datasets

We extensively compare our proposed technique to a series of baselines using the DAVIS dataset [42] (50 video sequences), the SegTrack v2 dataset [33] (14 video sequences), and the FBMS-59 dataset [39] (22 video sequences in the test set). These datasets are challenging as they contain nonrigid deformation, drastic illumination changes, cluttered background, rapid object motion, and occlusion. All three datasets provide pixel-level ground-truth annotations for each frame.

4.3 Evaluation metrics

Intersection over union ($\mathscr{J}$): The intersection over union (IoU) metric, also called the Jaccard index, computes the average over the dataset. The IoU metric has been widely used for evaluating the quality of the segmentation.
Contour accuracy ($\mathscr{F}$) [42]: To assess the segmentation quality, we compute the contour accuracy as $\mathscr{F} = \frac{2PR}{P+R}$, where P and R are the matching precision and recall of the

Table 1: Contribution of different components of our algorithm evaluated on the DAVIS dataset. Our algorithm with inter-frame, intra-frame connections, long range connections, and focused diffusion (denoted as FDiff) enabled performs best and achieves an IoU of **77.56**%.

Connections			FDiff	IoU (%)
Inter-frame	Intra-frame	Long range		
-	-	-	-	57.52
✓	-	-	-	62.75
-	✓	-	-	62.13
-	-	✓	-	72.38
✓	✓	-	-	65.01
✓	-	✓	-	72.70
-	✓	✓	-	74.13
✓	✓	✓	-	74.34
✓	✓	✓	✓	**77.56**

two sets of points on the contours of the ground truth segment and the output segment, calculated via a bipartite graph matching.

Temporal stability ($\mathscr{T}$) [42]: The temporal stability is measured by computing the distance between the shape context descriptors [3] describing the shape of the boundary of the segmentations between two successive frames. Intuitively, the metric indicates the degree of deformation required to transform the segmentation mask from one frame to its adjacent frames.

Subsequently we first present an ablation study where we assess the contributions of our technique. Afterwards we perform a quantitative evaluation where we compare the accuracy of our approach to baseline video segmentation approaches. Finally we present qualitative results to illustrate the success and failure cases of our method.

4.4 Ablation study

We assess the resulting performance of the individual components of our adjacency defined neighborhood in Table 1. The performance in IoU of the motion saliency estimation in our approach (with all the connections disabled) is 57.52%. We analyze the effect of the three main components in the adjacency graph: (1) inter-frame flow based temporal connections T, (2) intra-frame edge based spatial connections E and (3) long range connections V.

The improvements reported for saliency estimation and neighborhood construction motivate their use for unsupervised video segmentation. Besides, we apply a second round of 'focused diffusion,' restricted to the region which focuses primarily on the foreground object, to improve the results. The effects of the focused diffusion (denoted 'FDiff') can be found in Table 1 as well, showing significant improvements.

In Table 1, the checkmark '✓' indicates the *enabled* components. We observe consistent improvements when including additional components, which improve the robustness of the proposed method.

Table 2: The quantitative evaluation on the DAVIS dataset [42]. Evaluation metrics are the IoU measurement $\mathscr{J}$, boundary precision $\mathscr{F}$, and time stability $\mathscr{T}$. Following [42], we also report the recall and the decay of performance over time for $\mathscr{J}$ and $\mathscr{F}$ measurements.

		Semi-supervised										Unsupervised							
		SEA	HVS	JMP	FCP	BVS	OFL	CTN	VPN	MSK	**OURS-S**	NLC	MSG	KEY	FST	FSG	LMP	ARP	**OURS-U**
Deep features		-	-	-	-	-	✓	✓	✓	✓	-	-	-	-	-	✓	✓	-	-
$\mathscr{J}$	Mean $\mathscr{M}$ ↑	0.556	0.596	0.607	0.631	0.665	0.711	0.755	0.750	0.803	**0.810**	0.641	0.543	0.569	0.575	0.716	0.697	0.763	**0.776**
	Recall $\mathscr{O}$ ↑	0.606	0.698	0.693	0.778	0.764	0.800	0.890	0.901	0.935	**0.946**	0.731	0.636	0.671	0.652	0.877	0.829	**0.892**	0.886
	Decay $\mathscr{D}$ ↓	0.355	0.197	0.372	**0.031**	0.260	0.227	0.144	0.093	0.089	0.102	0.086	0.028	0.075	0.044	**0.017**	0.056	0.036	0.044
$\mathscr{F}$	Mean $\mathscr{M}$ ↑	0.533	0.576	0.586	0.546	0.656	0.679	0.714	0.724	0.758	**0.783**	0.593	0.525	0.503	0.536	0.658	0.663	0.711	**0.750**
	Recall $\mathscr{O}$ ↑	0.559	0.712	0.656	0.604	0.774	0.780	0.848	0.842	0.882	**0.928**	0.658	0.613	0.534	0.579	0.790	0.783	0.828	**0.869**
	Decay $\mathscr{D}$ ↓	0.339	0.202	0.373	**0.039**	0.236	0.240	0.140	0.136	0.095	0.115	0.086	0.057	0.079	0.065	0.043	0.067	0.073	**0.042**
$\mathscr{T}$	Mean $\mathscr{M}$ ↓	0.137	0.296	**0.131**	0.285	0.316	0.239	0.198	0.300	0.189	0.212	0.356	0.250	**0.190**	0.276	0.286	0.689	0.352	0.243

Table 3: The attribute-based aggregate performance comparing unsupervised methods on the DAVIS dataset [42]. We calculate the average IoU of all sequences with the specific attribute: appearance change (AC), dynamic background (DB), fast motion (FM), motion blur (MB), and occlusion (OCC). The right column with small font indicates the performance change for the method on the remaining sequences if the sequences possessing the corresponding attribute are not taken into account.

Attribute	NLC [13]	MSG [5]	KEY [31]	FST [40]	FSG [25]	LMP [49]	ARP [30]	**OURS-U**
AC	0.54 *+0.13*	0.48 *+0.08*	0.42 *+0.19*	0.55 *+0.04*	**0.73** ***-0.02***	0.67 *+0.03*	0.73 *+0.04*	0.72 *+0.07*
DB	0.53 *+0.15*	0.43 *+0.15*	0.52 *+0.07*	0.53 *+0.06*	0.67 *+0.05*	0.57 *+0.16*	**0.70** ***+0.08***	0.66 *+0.15*
FM	0.64 *+0.00*	0.46 *+0.14*	0.50 *+0.12*	0.50 *+0.12*	0.69 *+0.04*	0.67 *+0.05*	0.73 *+0.05*	**0.75** ***+0.04***
MB	0.61 *+0.04*	0.35 *+0.29*	0.51 *+0.08*	0.48 *+0.14*	0.65 *+0.10*	0.64 *+0.08*	0.69 *+0.11*	**0.74** ***+0.06***
OCC	0.70 *-0.09*	0.48 *+0.10*	0.52 *+0.08*	0.53 *+0.07*	0.65 *+0.10*	0.70 *-0.01*	0.71 *+0.08*	**0.81** ***-0.05***

4.5 Quantitative evaluation

Evaluation on the DAVIS dataset: We compare the performance of our approach to several baselines using the DAVIS dataset. The results are summarized in Table 2, where we report the IoU, the contour accuracy, and the time stability metrics. The best method is emphasized in bold font and the second best is underlined. We observe our approach to be quite competitive, outperforming a wide variety of existing unsupervised video segmentation techniques, *e.g.*, NLC [13], MSG [5], KEY [31], FST [40], FSG [25], LMP [49], ARP [30]. We also evaluate our method in the semi-supervised setting by simply replacing the saliency initialization of the first frame with the ground truth. Note that it is common to refer to usage of the first frame as 'semi-supervised.' Our unsupervised version is denoted as **OURS-U** and the semi-supervised version is referred to via **OURS-S** in Table 2. Semi-supervised baselines are SEA [1], HVS [17], JMP [14], FCP [43], BVS [36], OFL [52], CTN [27], VPN [26], and MSK [41]. Note that OFL uses deep features, and CTN, VPN, MSK, FSG, and LMP are deep learning based approaches. We observe our method to improve the state-of-the-art performance in IoU metric by 1.3% in the unsupervised setting and by 0.7% in the semi-supervised case. Note that beyond training of edge detectors, no learning is performed in our approach.

In Table 3, we compare the average IoU of all DAVIS sequences, clustered by attributes, *e.g.*, appearance change, dynamic blur, fast motion, motion blur, and occlusion.

Table 4: Performance in IoU on SegTrack v2 dataset [33].

Sequence	KEY [31]	FST [40]	NLC [13]	FSG [25]	Ours
BIRDFALL	0.490	0.014	0.565	0.380	**0.649**
BIRD OF PARADISE	0.922	0.837	0.814	0.699	**0.937**
BMX	0.630	0.621	0.754	0.591	**0.847**
CHEETAH	0.281	0.396	0.518	**0.596**	0.518
DRIFT	0.469	0.811	0.741	**0.876**	0.829
FROG	0.000	0.629	0.713	0.570	**0.832**
GIRL	**0.877**	0.441	0.860	0.667	0.846
HUMMINGBIRD	0.602	0.335	0.624	**0.652**	0.464
MONKEY	0.790	0.699	**0.823**	0.805	0.739
MONKEYDOG	0.396	0.523	**0.525**	0.328	0.381
PARACHUTE	**0.963**	0.839	0.859	0.516	0.937
PENGUIN	0.093	0.074	0.139	**0.713**	0.240
SOLDIER	0.666	0.453	0.692	0.698	**0.800**
WORM	**0.844**	0.705	0.782	0.506	0.800
Average IoU	0.573	0.527	0.672	0.614	**0.701**

Table 5: Performance in IoU on FBMS-59 test set [39].

	NLC [13]	POR [59]	POS [28]	FST [40]	ARP [30]	**OURS**
Average IoU	0.445	0.473	0.542	0.555	0.598	**0.608**

Our method is more robust and outperforms the baselines for fast motion, motion blur and occlusion. In particular, our method performs well for objects with occlusion, outperforming other methods by 10% for this attribute.

Evaluation on the SegTrack v2 dataset: We assess our approach on the SegTrack v2 dataset using identical choice of parameters. We show the results in Table 4. We observe our method to be competitive on SegTrack v2. Note that the reported performance of NLC differs from [13] as in the evaluation in [13] only a subset of the 12 video sequences were used. We ran the code released by [13] and report the results on the full SegTrack v2 dataset with 14 video sequences. The results we report here are similar to the ones reported in [48].

Evaluation on the FBMS dataset: We evaluate our method on the FBMS [39] test set which consists of 22 video sequences. The results are presented in Table 5. We observe our approach to outperform the baselines.

Comparisons of the saliency estimation: To illustrate the benefits of the proposed motion saliency estimation, we compare the performance of the proposed initialization with other approaches in Table 6 and observe that the proposed saliency estimation performs very well. Note that the saliency estimation in our approach is unsupervised as opposed to FSG and LMP which are trained on more than 10,000 images and 2,250 videos, respectively.

Table 6: Performance comparisons in IoU on the initialization on the DAVIS and Segtrack v2 datasets.

	DAVIS					Segtrack v2			
	NLC	FST	FSG	LMP	**Ours**	NLC	FST	FSG	**Ours**
Training?	-	-	✓	✓	-	-	-	✓	-
Initial saliency	0.402	0.456	**0.602**	0.569	0.575	0.419	0.389	**0.530**	0.424

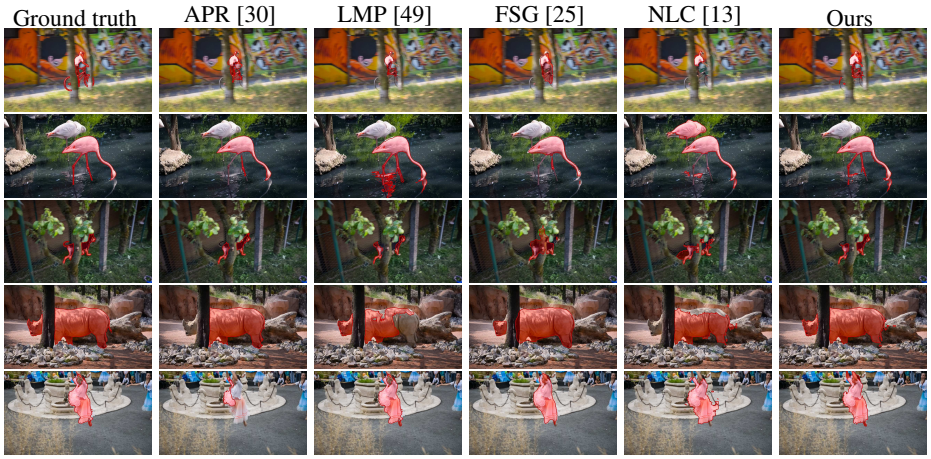

Fig. 4: **Comparison** of our algorithm and other unsupervised methods on sequence BMX-TREES (1st row), FLAMINGO (2nd row), LIBBY (3rd row), RHINO (4th row), and DANCE-JUMP (5th row) of the DAVIS dataset.

4.6 Qualitative evaluation

Side-by-side comparison: Next we present qualitative results comparing our algorithm to competing methods on challenging parts of the DAVIS dataset. In Figure 4 we provide side-by-side comparisons to existing methods, *i.e.*, APR [30], LMP [49], FSG [25], and NLC [13]. We observe our approach to yield encouraging results even in challenging situations such as frames in BMX-TREES (Figure 4, first row), where the foreground object is very small and occluded, and the background is very colorful, and in FLAMINGO (Figure 4, second row), where there is non-rigid deformation, and the background object is similar to the foreground object. We refer the interested reader to the supplementary material for additional results and videos.

Success cases: In Figure 5, we provide success cases of our algorithm, *i.e.*, frames where our designed technique delineates the foreground object accurately. We want to highlight that our approach is more robust to challenges such as occlusions, motion blur and fast moving objects as the attribute-based aggregate performance in Table 3 suggests.

Fig. 5: **Visual results** of our approach on the sequences SWING (1st row), SOAPBOX (2nd row), DRIFT-STRAIGHT (3rd row), and DANCE-TWIRL (4th row) of the DAVIS dataset.

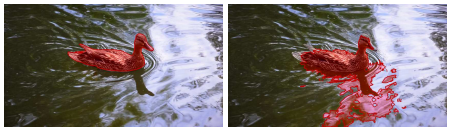

Fig. 6: **Failure case.** Groundtruth vs. our result.

Failure modes: In Figure 6, we also present failure modes of our approach. We observe our technique to be challenged by complex motion. Since our method mainly relies on motion and appearance, water is classified as foreground due to its complex motion (MALLARD-WATER).

5 Conclusion

We proposed a saliency estimation and a graph neighborhood for effective unsupervised foreground-background video segmentation. Our key novelty is a motion saliency estimation and an informative neighborhood structure. Our unsupervised method demonstrates how to effectively exploit the structure of video data, *i.e.*, taking advantage of flow and edges, and achieves state-of-the-art performance in the unsupervised setting.

Acknowledgments: This material is based upon work supported in part by the National Science Foundation under Grant No. 1718221, 1755785, Samsung, and 3M. We thank NVIDIA for providing the GPUs used for this research.

References

1. Avinash Ramakanth, S., Venkatesh Babu, R.: SeamSeg: Video object segmentation using patch seams. In: Proc. CVPR (2014)
2. Badrinarayanan, V., Galasso, F., Cipolla, R.: Label propagation in video sequences. In: Proc. CVPR (2010)
3. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. PAMI (2002)
4. Brendel, W., Todorovic, S.: Video object segmentation by tracking regions. In: Proc. ICCV (2009)
5. Brox, T., Malik, J.: Object segmentation by long term analysis of point trajectories. In: Proc. ECCV (2010)
6. Brutzer, S., Hoeferlin, B., Heidemann, G.: Evaluation of background subtraction techniques for video surveillance. In: Proc. CVPR (2011)
7. Cheng, H.T., Ahuja, N.: Exploiting nonlocal spatiotemporal structure for video segmentation. In: Proc. CVPR (2012)
8. Costeira, J., Kanande, T.: A multi-body factorization method for motion analysis. In: Proc. ICCV (1995)
9. Criminisi, A., Cross, G., Blake, A., Kolmogorov, V.: Bilayer segmentation of live video. In: Proc. CVPR (2006)
10. Dollár, P., Zitnick, C.L.: Structured forests for fast edge detection. In: ICCV (2013)
11. Elgammal, A., Duraiswami, R., Harwood, D., Davis, L.: Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. In: Proc. IEEE (2002)
12. Elhamifar, E., Vidal, R.: Sparse subspace clustering. In: Proc. CVPR (2009)
13. Faktor, A., Irani, M.: Video segmentation by non-local consensus voting. In: BMVC (2014)
14. Fan, Q., Zhong, F., Lischinski, D., Cohen-Or, D., Chen, B.: JumpCut: Non-successive mask transfer and interpolation for video cutout. SIGGRAPH (2015)
15. Fragkiadaki, K., Zhang, G., Shi, J.: Video segmentation by tracing discontinuities in a trajectory embedding. In: Proc. CVPR (2012)
16. Galasso, F., Nagaraja, N., Cardenas, T., Brox, T., Schiele, B.: A unified video segmentation benchmark: Annotation, metrics and analysis. In: Proc. ICCV (2013)
17. Grundmann, M., Kwatra, V., Han, M., Essa, I.: Efficient hierarchical graph-based video segmentation. In: Proc. CVPR (2010)
18. Haymanand, E., Eklundh, J.O.: Statistical background subtraction for a mobile observer. In: Proc. ICCV (2003)
19. Hu, Y.T., Huang, J.B., Schwing, A.G.: MaskRNN: Instance Level Video Object Segmentation. In: Proc. NIPS (2017)
20. Hu, Y.T., Huang, J.B., Schwing, A.G.: VideoMatch: Matching based Video Object Segmentation. In: Proc. ECCV (2018)
21. Hu, Y., Song, R., Li, Y.: Efficient coarse-to-fine patchmatch for large displacement optical flow. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
22. Irani, M., Anandan, P.: A unified approach to moving object detection in 2d and 3d scenes. PAMI (1998)
23. Irani, M., Rousso, B., Peleg, S.: Computing occluding and transparent motions. IJCV (1994)
24. Jain, S.D., Grauman, K.: Supervoxel-consistent foreground propagation in video. In: Proc. ECCV (2014)
25. Jain, S.D., Xiong, B., Grauman, K.: FusionSeg: Learning to combine motion and appearance for fully automatic segmention of generic objects in videos. Proc. CVPR (2017)

26. Jampani, V., Gadde, R., Gehler, P.V.: Video propagation networks. In: Proc. CVPR (2017)
27. Jang, W.D., Kim, C.S.: Online video object segmentation via convolutional trident network. In: Proc. CVPR (2017)
28. Jang, W.D., Lee, C., Kim, C.S.: Primary object segmentation in videos via alternate convex optimization of foreground and background distributions. In: Proc. CVPR (2016)
29. Keuper, M., Andres, B., Brox, T.: Motion trajectory segmentation via minimum cost multicuts. In: Proc. ICCV (2015)
30. Koh, Y.J., Kim, C.S.: Primary object segmentation in videos based on region augmentation and reduction. In: Proc. CVPR (2017)
31. Lee, Y.J., Kim, J., Grauman, K.: Key-segments for video object segmentation. In: Proc. ICCV (2011)
32. Lezama, J., Alahari, K., Sivic, J., Laptev, I.: Track to the future: Spatio-temporal video segmentation with long-range motion cues. In: Proc. CVPR (2011)
33. Li, F., Kim, T., Humayun, A., Tsai, D., Rehg, J.M.: Video segmentation by tracking many figure-ground segments. In: Proc. ICCV (2013)
34. Li, W., Viola, F., Starck, J., Brostow, G.J., Campbell, N.D.: Roto++: Accelerating professional rotoscoping using shape manifolds. SIGGRAPH (2016)
35. Lovsz, L.: Random walks on graphs: A survey (1993)
36. Maerki, N., Perazzi, F., Wang, O., Sorkine-Hornung, A.: Bilateral space video segmentation. In: Proc. CVPR (2016)
37. Muja, M., Lowe, D.G.: Fast approximate nearest neighbors with automatic algorithm configuration. In: International Conference on Computer Vision Theory and Application (2009)
38. Nagaraja, N., Schmidt, F., Brox, T.: Video segmentation with just a few strokes. In: Proc. ICCV (2015)
39. Ochs, P., Malik, J., Brox, T.: Segmentation of moving objects by long term video analysis. PAMI (2014)
40. Papazoglou, A., Ferrari, V.: Fast object segmentation in unconstrained video. In: Proc. ICCV (2013)
41. Perazzi, F., Khoreva, A., Benenson, R., Schiele, B., A.Sorkine-Hornung: Learning video object segmentation from static images. In: Proc. CVPR (2017)
42. Perazzi, F., Pont-Tuset, J., McWilliams, B., Gool, L.V., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: Proc. CVPR (2016)
43. Perazzi, F., Wang, O., Gross, M., Sorkine-Hornung, A.: Fully connected object proposals for video segmentation. In: Proc. ICCV (2015)
44. Price, B.L., Morse, B.S., Cohen, S.: LIVEcut: Learning-based interactive video segmentation by evaluation of multiple propagated cues. In: Proc. ICCV (2009)
45. Rao, S.R., Tron, R., Vidal, R., Ma, Y.: Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories. In: Proc. CVPR (2008)
46. Ren, Y., Chua, C.S., Ho, Y.K.: Statistical background modeling for non-stationary camera. PRL (2003)
47. Revaud, J., Weinzaepfel, P., Harchaoui, Z., Schmid, C.: EpicFlow: Edge-Preserving Interpolation of Correspondences for Optical Flow. In: Computer Vision and Pattern Recognition (2015)
48. Tokmakov, P., Alahari, K., Schmid, C.: Learning video object segmentation with visual memory. In: Proc. ICCV (2017)
49. Tokmakov, P., Alahari, K., Schmid, C.: Learning motion patterns in videos. In: Proc. CVPR (2017)
50. Torr, P.H.S., Zisserman, A.: Concerning bayesian motion segmentation, model averaging, matching and the trifocal tensor. In: Proc. ECCV (1998)

51. Tsai, D., Flagg, M., Rehg, J.: Motion coherent tracking with multi-label mrf optimization. In: Proc. BMVC (2010)
52. Tsai, Y.H., Yang, M.H., Black, M.J.: Video Segmentation via Object Flow. In: Proc. CVPR (2016)
53. Tu, W.C., He, S., Yang, Q., Chien, S.Y.: Real-time salient object detection with a minimum spanning tree. In: Proc. CVPR (2016)
54. Vijayanarasimhan, S., Grauman, K.: Active frame selection for label propagation in videos. In: Proc. ECCV (2012)
55. Wang, T., Collomosse, J.: Probabilistic motion diffusion of labeling priors for coherent video segmentation. IEEE Transactions on Multimedia (2012)
56. Wei, Y., Wen, F., Zhu, W., Sun, J.: Geodesic saliency using background priors. In: Proc. ECCV (2012)
57. Xiao, F., Lee, Y.J.: Track and segment: An iterative unsupervised approach for video object proposals. In: Proc. CVPR (2016)
58. Yuan, C., Medioni, G., Kang, J., Cohen, I.: Detecting motion regions in the presence of a strong parallax from a moving camera by multiview geometric constraints. PAMI (2007)
59. Zhang, D., Javed, O., Shah, M.: Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In: Proc. CVPR (2013)