Super-Trajectory for Video Segmentation

Wenguan Wang [1], Jianbing Shen* [1], Jianwen Xie [2], and Fatih Porikli [3]

[1]Beijing Lab of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, China
[2]University of California, Los Angeles, USA
[3]Research School of Engineering, Australian National University, Australia

Abstract

We introduce a novel semi-supervised video segmentation approach based on an efficient video representation, called as "super-trajectory". Each super-trajectory corresponds to a group of compact trajectories that exhibit consistent motion patterns, similar appearance and close spatiotemporal relationships. We generate trajectories using a probabilistic model, which handles occlusions and drifts in a robust and natural way. To reliably group trajectories, we adopt a modified version of the density peaks based clustering algorithm that allows capturing rich spatiotemporal relations among trajectories in the clustering process. The presented video representation is discriminative enough to accurately propagate the initial annotations in the first frame onto the remaining video frames. Extensive experimental analysis on challenging benchmarks demonstrate our method is capable of distinguishing the target objects from complex backgrounds and even reidentifying them after occlusions.

1. Introduction

We state the problem of semi-supervised video object segmentation as the partitioning of objects in a given video sequence with available annotations in the first frame. Aiming for this task, we incorporate an efficient video representation, *super-trajectory*, to capture the underlying spatiotemporal structure information that is intrinsic to real-word scenes. Each super-trajectory corresponds to a group of trajectories that are similar in nature and have common characteristics. A point trajectory, *e.g.*, the tracked positions of an individual point across multiple frames, is a constituent of super-trajectory. This representation captures several aspects of a video:

- *Long-term motion information* is explicitly modeled as it consists of trajectories over extended periods;
- *Spatiotemporal location information* is implicitly interpreted by clustering nearby trajectories; and
- *Compact features*, such as color and motion pattern, are described in a conveniently compact form.

With above good properties, super-trajectory simplifies and reduces the complexity of propagating human-provided labels in the segmentation process. We first generate trajectory based on Markovian process, which handles occlusions and drifts naturally and efficiently. Then a density peaks based clustering (DPC) algorithm [31] is modified for obtaining reasonable division of the trajectories, which offers proper split of videos in space and time axes. The design of our super-trajectory is motivated by the flowing two aspects.

Figure 1. Our video segmentation method takes the first frame annotation as initialization (left). Leveraging on super-trajectories, the segmentation process achieves superior results even for challenging scenarios including heavy occlusions, complex appearance variations, and large shape deformations (middle, right).

*Corresponding author: Jianbing Shen (shenjianbing@bit.edu.cn).
This work was supported in part by the National Basic Research Program of China (973 Program) (No. 2013CB328805), the National Natural Science Foundation of China (61272359), the Australian Research Council's Discovery Projects funding scheme (project DP150104645), and the Fok Ying-Tong Education Foundation for Young Teachers. Specialized Fund for Joint Building Program of Beijing Municipal Education Commission.

Firstly, for the task of video segmentation, it is desirable to have a powerful abstraction of videos that is robust to structure variations and deformations in image space and time. As demonstrated in recently released DAVIS dataset [28], most of the existing approaches exhibit severe limitations for occlusions, motion blur, and appearance changes. The proposed super-trajectory, encoded with several well properties, is able to capture above instances (see Fig. 1).

Secondly, from the perspective of feature generation, point trajectory is desired to be improved for meeting above requests. Merging and splitting video segments (and corresponding trajectories) into atomic spatiotemporal components is essential for handling occlusions and temporal discontinuities. However, it is well-known that, classical clustering methods (*e.g.*, k-means and spectral clustering), which are widely adopted by previous trajectory methods, even cannot reached a consensus on the definition of a cluster. Here, we modify DPC algorithm [31] for grouping trajectories, favoring its advantage of choosing cluster centers based on a reasonable criterion.

We conduct video segmentation via operating trajectories as unified super-trajectory groups. To eliminate adverse effects of camera motion, we introduce a *reverse-tracking* strategy to exclude objects that originate outside the frame. To reidentify objects after occlusions, we exploit *object reoccurrence* information, which reflects the spatiotemporal relations of objects across the entire video sequence.

The remainder of the article is organized as follows. A summarization of related work is introduced in Sec. 2. Our approach for super-trajectory generation is presented in detail in Sec. 3. In Sec. 4, we describe our super-trajectory based video segmentation algorithm. We then experimentally demonstrate its robustness, effectiveness, and efficiency in Sec. 5. Finally, we draw conclusions in Sec. 6.

2. Related Work

We provide a brief overview of recent works in video object segmentation and point trajectory extraction.

2.1. Video Object Segmentation

According to the level of supervision required, video segmentation techniques can be broadly categorized as unsupervised, semi-supervised and supervised methods.

Unsupervised algorithms [37, 49, 26, 44, 42] do not require manual annotations but often rely on certain limiting assumptions about the application scenario. Some techniques [5, 14, 27] emphasize the importance of motion information. More specially, [5, 14] analyze long-term motion information via trajectories, then solve the segmentation as a trajectory clustering problem. The works [10, 43, 46] introduce saliency information [45] as prior knowledge to infer the object. Recently, [19, 22, 50, 12, 48] generate object segments via ranking several object candidates.

Semi-supervised video segmentation, which also refers to *label propagation*, is usually achieved via propagating human annotation specified on one or a few key-frames onto the entire video sequence [4, 2, 35, 7, 21, 17, 32, 36]. These methods mainly use flow-based random field propagation models [38], patch-seams based propagation strategies [30], energy optimizations over graph models [29], joint segmentation and detection frameworks [47], or pixel segmentation on bilateral space [23].

Supervised methods [3, 51, 11] require tedious user interaction and iterative human corrections. These methods can attain high-quality boundaries while suffering from extensive and time-consuming human supervision.

2.2. Point Trajectory

Point trajectories are generated through tracking points over multiple frames and have the advantage of representing long-term motion information. Kanade-Lucas-Tomasi (KLT) [33] is among the most popular methods that track a small amount of feature points. Inspiring several follow-up studies in video segmentation and action recognition, optical flow based dense trajectories [34] improve over sparse interest point tracking. In particular, [39, 40, 41] introduce dense trajectories for action recognition. Other methods [5, 13, 20, 15, 14, 24, 25, 18, 9] address the problem of unsupervised video segmentation, in which case the problem also be described as *motion segmentation*. These methods usually track points via dense optical flow and perform segmentation via clustering trajectories.

Existing approaches often handle trajectories in pairs or individually and directly group all the trajectories into few clusters as segments, easily ignoring the inner coherence in a group of similar trajectories. Instead, we operate trajectories as united super-trajectory groups instead of individual entities, thus offering compact and atomic video representation and fully exploiting spatiotemporal relations among trajectories.

3. Super-Trajectory via Grouping Trajectories

3.1. Trajectory Generation

Given a sequence of video frames $I_{1:T} = \{I_1, \cdots, I_T\}$ within time range $[1, T]$, each pixel point can be tracked to the next frame using optical flow. This tracking process can be executed frame-by-frame until some termination conditions (*e.g.*, occlusion, incorrect motion estimates, *etc.*) are reached. The tracked points are composed into a trajectory and a new tracker is initialized where prior tracker finished. We build our trajectory generation on a unified probabilistic model which naturally considers various termination conditions.

Let $\mathbf{w}$ denote a flow field indexed by pixel positions that returns a 2D flow vector at a given point. Using LDOF [6], we compute forward-flow field $\mathbf{w}_t$ from frame I_t to I_{t+1}, and the backward-flow field $\hat{\mathbf{w}}_t$ from I_t to I_{t-1}. We track pixel potion $\mathbf{x} = (x, y, t)$ to the consecutive frames in both directions. The tracked points of consecutive frames are concatenated to form a trajectory τ:

$$\tau = \{\mathbf{x}_n\}_{n=1}^{L} = \{(x_n, y_n, t_n)\}_{n=1}^{L}, \quad t_n \in [1, T], \quad (1)$$

where L indicates the length of trajectory τ and $(x_n, y_n) = (x_{n-1}, y_{n-1}) + \mathbf{w}_{t_{n-1}}(x_{n-1}, y_{n-1})$. We model point tracking process as a first order Markovian process, and denote the probability that n-th point $\mathbf{x}_n$ of trajectory τ is correctly tracked from frame I_{t_1} as $p(\mathbf{x}_n|I_{t_1:t_n})$. The prediction model is defined by:

$$p(\mathbf{x}_n|I_{t_1:t_n}) = p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{n-1}|I_{t_1:t_{n-1}}), \quad (2)$$

where $p(\mathbf{x}_1|I_{t_1}) = 1$ and $p(\mathbf{x}_n|\mathbf{x}_{n-1})$ is formulated as:

$$p(\mathbf{x}_n|\mathbf{x}_{n-1}) = exp\{-(\mathcal{E}_{app} + \mathcal{E}_{occ})\}. \quad (3)$$

The energy functions $\mathcal{E}$ penalize various potential tracking error. The former energy $\mathcal{E}_{app}$ is expressed as:

$$\mathcal{E}_{app}(\mathbf{x}_n, \mathbf{x}_{n-1}) = ||I_{t_n}(x_n, y_n) - I_{t_{n-1}}(x_{n-1}, y_{n-1})||, \quad (4)$$

which penalizes the appearance variations between corresponding points. The latter energy $\mathcal{E}_{occ}$ is included to penalize occlusions. It uses the consistency of the forward and backward flows:

$$\mathcal{E}_{occ}(\mathbf{x}_n, \mathbf{x}_{n-1}) = \frac{||\hat{\mathbf{w}}_{t_n}(x_n, y_n) + \mathbf{w}_{t_{n-1}}(x_{n-1}, y_{n-1})||}{||\hat{\mathbf{w}}_{t_n}(x_n, y_n)|| + ||\mathbf{w}_{t_{n-1}}(x_{n-1}, y_{n-1})||}. \quad (5)$$

When this consistency constraint is violated, occlusions or unreliable optical flow estimates might occur (see [34] for more discussion). It is important to notice that the proposed tracking model performs accurately yet our model is not limited to the above constraints. We terminate the tracking process when $p(\mathbf{x}_n|I_{t_1:t_n}) < 0.5$, and then we start a new tracker at $\mathbf{x}_n$. In our implementation, we discard the trajectories shorter than four frames.

3.2. Super-Trajectory Generation

Previous studies indicate the value of trajectory based representation for long-term motion information. Our additional intuition is that neighbouring trajectories exhibit compact spatiotemporal relationships and they have similar natures in appearance and motion patterns. This motives us operating on trajectories as united groups.

We generate super-trajectory by clustering trajectories with density peaks based clustering (DPC) algorithm[31]. Before introducing our super-trajectory generation method, we first describe DPC.

Density Peaks based Clustering (DPC) DPC is proposed to cluster the data by finding of density peaks. It provides a unique solution of fast clustering based on the idea that cluster centers are characterized by a higher density than their neighbors and by a relatively large distance from points with higher densities. It offers a reasonable criterion for finding clustering centers.

Given the distances d_{ij} between data points, for each data point i, DPC calculates two quantities: local density ρ_i and its distance δ_i from points of higher density. The local density ρ_i of data point i is defined as [1]:

$$\rho_i = \sum\nolimits_j d_{ij}. \quad (6)$$

Here, δ_i is measured by computing the minimum distance between the point i and any other point with higher density:

$$\delta_i = \min_{j:\rho_j > \rho_i} (d_{ij}). \quad (7)$$

For the point with highest density, it takes $\delta_i = max_j(d_{ij})$.

Cluster centers are the points with high local density ($\rho \uparrow$) and large distance ($\delta \uparrow$) from other points with higher local density. The data points can be ranked via $\gamma_i = \rho_i \delta_i$, and the top ranking points are selected as centers. After successfully declaring cluster centers, each remaining data points is assigned to the cluster center as its nearest neighbor of higher density.

Grouping Trajectories via DPC Given a trajectory τ : $\{(x_n, y_n, t_n)\}_n$ spans L frames, we define three features: spatial location (l_τ), color (c_τ), and velocity (v_τ), for describing τ:

$$\begin{aligned} l_\tau &= \frac{1}{L}\sum_{n=1}^{L}(x_n, y_n), \;\; c_\tau = \frac{1}{L}\sum_{n=1}^{L} I_{t_n}(x_n, y_n), \\ v_\tau &= \frac{1}{L}\sum_{n=1}^{L}\big(\frac{1}{\Delta t}(x_{n+\Delta t} - x_n, y_{n+\Delta t} - y_n)\big), \end{aligned} \quad (8)$$

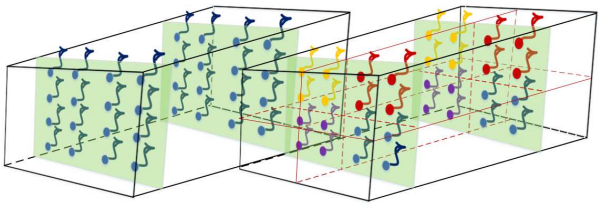
where we set $\Delta t = 3$. We tested $\Delta t = \{5, 7, 9\}$ and did not observe obvious effect on the results.

Between each pair of trajectories τ_i and τ_j that share some frames, we define their distance d_{ij} via measuring descriptor similarity:

$$d_{ij} = \sum\nolimits_{f \in \{l,c,v\}} ||f_{\tau_i} - f_{\tau_j}||. \quad (9)$$

We normalize color distance on max intensity, location distance on sampling step R (detailed below), motion distance on the mean motion magnitude of all the trajectories, which makes above distance measures to have similar scales. In case there is no temporal overlap, we set $d_{ij} = H$, where H has a very large value.

[1] Here we do not use the cut-off kernel or gaussian kernel adopted in [31], due to the small data amount.



(a) (b)

Figure 2. Illustration of initial super-trajectory generation. (a) The arrows indicate trajectories and the dots indicate the initial location of trajectory. (b) We roughly divide all the trajectories into K groups with a given number of spatial grids $K = 4$.

We first roughly partition trajectories into several non-overlap clusters, and then iteratively updates each partition to get the optimized trajectory clusters.

The only parameter of our super-trajectory algorithm is number of spatial grids K, as the degree of spatial subdivision. The spatial sampling step becomes $R = \sqrt{S/K}$, where S refers to the product of the height and width of image frame. The clustering procedure begins with an initialization step where we divide the input video $I_{1:T}$ into several non-overlap spatiotemporal volumes of size $R \times R \times T$. As shown in Fig. 2, all trajectories $\mathcal{T} = \{\tau_i\}_i$ are divided into K volumes. A trajectory τ falls into the volume where it starts. Then we need to find a proper cluster number of each trajectory group, thereby further offering a reasonable temporal split of video.

For each trajectory group, we initially estimate the cluster number as $C = T/\overline{L}$, where $\overline{L}$ indicates the average length of all the trajectories. Then we apply a modified DPC algorithm for generating trajectory clusters, as described with in Alg. 1. In Alg. 1-3, if we have $\delta_i = H$, then trajectory τ_i does not have any temporal overlap with those trajectories have higher local densities. That means trajectory τ_i is the center of a isolated group. If $C < n'$, in Alg. 1-4, that means there exist more than C unconnected trajectory groups. Then we select the trajectories with highest densities of those unconnected trajectory groups as centers (Alg. 1-5). Otherwise, in Alg. 1-7,8, the trajectories with the C highest γ values are selected as the cluster centers. The whole initialization process is described in Alg. 2-1,2,3.

Algorithm 1 DPC for Generating Super-Trajectory Centers

Input: A sub-group of trajectories $\mathcal{T}' = \{\tau'_i\}_i$ ($\mathcal{T}' \subset \mathcal{T}$), distance matrix $\{d_{ij}\}$ via Eq. 9 and cluster number C;
Output: Organized trajectory clusters;

1: Compute local densities $\{\rho_i\}_i$ via Eq. 6;
2: Compute distance $\{\delta_i\}_i$ via Eq. 7;
3: Find $\{\tau'_{i'}\}_{i'}$ with $\delta_{i'} = H$, where $|\{\tau'_{i'}\}_{i'}| = n'$;
4: **if** $C < n'$ **then**
5: Select $\{\tau'_{i'}\}_{i'}$ as cluster centers;
6: **else**
7: Compute $\{\gamma_i\}_i$ via $\gamma_i = \rho_i \delta_i$;
8: Select the trajectories with C highest γ values as cluster centers;
9: **end if**
10: Assign remaining trajectories to cluster centers.

(a) Frame It (b) 1st Iteration (c) 2nd Iteration (d) 3rd Iteration (e) 4th Iteration (f) 5th Iteration

Figure 3. Illustration of our super-trajectory generation via iterative trajectory clustering. (a) Frame I_t. (b)-(f) Visualization results of super-trajectory in time slice I_t with different iterations. Each pixel is assigned the average color of all the points along the trajectory which it belongs to. The blank areas are the discarded trajectories which are shorter than four frames.

Based on the above initialization process, we group trajectories into super-trajectories according to their spatiotemporal relationships and similarities (see Fig. 3(b)). Next, we iteratively refine our super-trajectory assignments. In this process, each trajectory is classified into the nearest cluster center. For reducing the searching space, we only search the trajectories fall into a $2R \times 2R \times T$ space-time volume around the cluster center $\tau_{i'}$ (Alg. 2-7). This results in a significant speed advantage by limiting the size of search space to reduce the number of distance calculations. Once each trajectory has been associated to the nearest cluster center, an update step adjusts the center of each trajectory cluster via Alg. 1 with $C = 1$ (Alg. 2-14,15). We drop very small trajectory clusters and combine those trajectories to other nearest trajectory clusters. In practice, we find 5 iterations for above refining process are enough for obtaining satisfactory performance. Visualization results of super-trajectory generation with different iterations are presented in Fig. 3.

Using Alg. 1, we group all trajectories $\mathcal{T} = \{\tau_i\}_i$ into m nonoverlap clusters, represented as super-trajectories $\mathcal{X} = \{\chi_j\}_{j=1}^m$, where $\chi_j = \{\tau_i \mid \tau_i$ is classified into j-th cluster

via Alg. 2}. It is worth to note that, m (the number of super-trajectories) is varied in each iteration in Alg. 2 since we merge small clusters into other clusters. Additionally, m for different videos is different even with same input parameter K. That is important, since different videos have different temporal characteristics, thus we only constrain their spatial shape via K.

Algorithm 2 Super-Trajectory Generation

Input: All the trajectories $\{\tau_i\}_i$, spatial sampling step R;
Output: Super-trajectory assignments;

/* Initialization */
Obtain K trajectory groups via spatial sampling step R;
Set initial cluster number $C = T/\overline{L}$ for each group;
Obtain initial cluster centers $\{\tau_{i'}\}_{i'}$ from each trajectory group via Alg. 1, where $|\{\tau_{i'}\}_{i'}| = m$;
loop
  /* Iterative Assignment */
  Set label $l_i = -1$ and distance $\kappa_i = H$ for each trajectory τ_i;
  for each trajectory cluster center $\tau_{i'}$ **do**
    for each trajectory τ_j falls in a $2R \times 2R \times T$ space-time volume around $\tau_{i'}$ **do**
      Compute distance $d_{ji'}$ between τ_j and $\tau_{i'}$ via Eq. 9;
      if $d_{ji'} < \kappa_j$ **then**
        Set $\kappa_j = d_{ji'}$, $l_j = i'$;
      end if
    end for
  end for
  /* Update Assignment */
  Set cluster number $C = 1$ for each group;
  Update $\{\tau_{i'}\}_{i'}$ for each cluster via Alg. 1.
end loop

4. Super-Trajectory based Video Segmentation

In Sec. 3, we cluster a set of compact trajectories into super-trajectory. In this section, we describe our video segmentation approach that leverages on super-trajectories.

Given the mask $\mathcal{M}$ of the first frame, we seek a binary partitioning of pixels into foreground and background classes. Clearly, the annotation can be propagated to the rest of the video, using the trajectories that start at the first frame. However, only a few of points can be successfully tracked across the whole scene, due to occlusion, drift or unreliable motion estimation. Benefiting from our efficient trajectory clustering approach, super-trajectories are able to spread more annotation information over longer periods. This inspires us to base our label propagation process on super-trajectory.

For inferring the foreground probability of super-trajectories $\mathcal{X}$, we first divide all the trajectories $\mathcal{T}$ into three categories: foreground trajectories $\mathcal{T}^f$, background trajectories $\mathcal{T}^b$ and unlabeled trajectories $\mathcal{T}^u$, where $\mathcal{T} = \mathcal{T}^f \cup \mathcal{T}^b \cup \mathcal{T}^u$. The $\mathcal{T}^f$ and $\mathcal{T}^b$ are the trajectories which start at the first frame and are labeled by the annotation mask $\mathcal{M}$, while the $\mathcal{T}^u$ are the trajectories start at any frames except the first frame, thus cannot be labeled via $\mathcal{M}$. Accordingly, super-trajectories $\mathcal{X}$ are classified into two categories: labeled ones $\mathcal{X}^l$ and unlabeled ones $\mathcal{X}^u$. A labeled super-trajectory $\chi_j^l \in \mathcal{X}^l$ contains at least one labeled trajectory from $\mathcal{T}^f$ or $\mathcal{T}^b$, and its foreground probability can be computed as the ratio between the included foreground trajectories and the labeled ones it contains:

$$p_f(\chi_j^l) = \frac{|\chi_j^l \cap \mathcal{T}^f|}{|\chi_j^l \cap \mathcal{T}^f| + |\chi_j^l \cap \mathcal{T}^b|}. \tag{10}$$

For the points belonging to the labeled super-trajectory χ_j^l, their foreground probabilities are set as $p_f(\chi_j^l)$.

Then we build an appearance model for estimating the foreground probabilities of unlabeled pixels. The appearance model is built upon the labeled super-trajectories $\mathcal{X}^l$, consists of two weighted Gaussian Mixture Models over RGB colour values, one for the foreground and one for the background. The foreground GMM is estimated form all labeled super-trajectories $\mathcal{X}^l$, weighted by their foreground probabilities $\{p_f(\chi_j^l)\}_j$. The estimation of background GMM is analogous, with the weight replaced by the background probabilities $\{1-p_f(\chi_j^l)\}_j$. The appearance models leverage the foreground and background super-trajectories over many frames, instead of only using the first frame or labeled trajectories, therefore they can robustly estimate appearance information.

Although above model successfully propagates more annotation information across the whole video sequence, it still suffers from some difficulties: the model will be confused when a new object come into view (see Fig. 4 (b)). To this, we propose to *reverse track points* for excluding new incoming objects. We compute the 'source' of unlabeled trajectory $\tau_i^u \in \mathcal{T}^u$:

$$(x_0, y_0) = (x_1, y_1) - v_{\tau_i^u}, \tag{11}$$

where (x_1, y_1) indicates starting position and $v_{\tau_i^u}$ refers to velocity via Eq. 8. It is clear that, if the virtual position (x_0, y_0) is out of image frame domain, trajectory τ_i^u is a latecomer. For those trajectories $\mathcal{T}^o \subset \mathcal{T}^u$ start outside view, we treat them as background. Labeled super-trajectory $\chi_j^l \in \mathcal{X}^l$ is redefined as the one contains at least one trajectory from $\mathcal{T}^f$, $\mathcal{T}^b$ or $\mathcal{T}^o$, and Eq. 10 is updated as

$$p_f(\chi_j^l) = \frac{|\chi_j^l \cap \mathcal{T}^f|}{|\chi_j^l \cap \mathcal{T}^f| + |\chi_j^l \cap \mathcal{T}^b| + |\chi_j^l \cap \mathcal{T}^o|}. \tag{12}$$

Those outside trajectories $\mathcal{T}^o$ are also adopted for training appearance model in prior step. According to our exper-

(a) (b) (c) (d) (e)

Figure 4. (a) Input frames. (b) Foreground estimates via Eq. 10. (c) Foreground estimates via our reverse tracking strategy (Eq. 12). (d) Foreground estimates via backward re-occurrence based optimization (Eq. 14). (e) Final segmentation results.

iment, this assumption offers about 6% performance improvement. Foreground estimation results via our reverse tracking strategy are presented in Fig. 4 (c).

For re-identifying objects after long-term occlusions and constraining segmentation consistency, we explore *re-occurrence* of objects. As suggested by [10], objects, or regions, often re-occur both in space and in time. Here, we build correspondences among re-occurring regions across distant frames and transport foreground estimates globally. This process is based on super-pixel level, since super-trajectories cannot cover all of pixels.

Let $\{r_i\}_i$ be the superpixel set of input video. For each region, we search its N Nearest Neighbors (NNs) as its re-occurring regions using KD-tree search. For region r_i of frame I_t, we only search its NNs in previous frames $\{I_1, \cdots, I_t\}$. Such *backward search* strategy is for biasing the segmentation results of prior frames as the propagation accuracy degrades over time. Following [10], each region r_i is represented as a concatenation of several descriptors f_{r_i}: RGB and LAB color histograms (6 channels$\times$20 bins), HOG descriptor (9 cells$\times$6 orientation bins) computed over a 15×15 patch around superpixel center, and spatial coordinate of superpixel center. The spatial coordinate is with respect to image center and normalized into $[0, 1]$, which implicitly incorporates spatial consistency in NN-search.

After NN-search in the feature space, we construct a weight matrix W for all the regions $\{r_i\}_i$:

$$W_{ij} = \begin{cases} e^{-||f_{r_i} - f_{r_j}||} & \text{if } r_j \text{ is one of NNs of } r_i \\ 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

Then a probability transition matrix P is built via row-wise normalization of W. We define a column vector v that gathers all the foreground probabilities of $\{r_i\}_i$. The foreground probability of a superpixel is assigned as the average foreground probabilities of its pixels.

We iteratively update v via the probability transition matrix P. In each iteration k, we update $v^{(k)}$ via:

$$v^{(k)} = Pv^{(k-1)}, \quad (14)$$

which equivalents to updating foreground probability of a region with the weighted average of its NNs. In each iteration, we keep the foreground probabilities of those points belonging to labeled trajectories unchanged. Then we recompute $v^{(k)}$ and update it in next iteration. In this way, the relatively accurate annotation information of the labeled trajectories is preserved. Additionally, the annotation information is progressively propagated in a forward way and the super-trajectories based foreground estimates are consistent even across many distant frames (see Fig. 4 (d)).

After 10 iterations, the pixels (regions) with foreground probabilities lager than 0.5 are classified as foreground, thus obtaining final binary segments. In Sec. 5.2, we test $N = \{4, 6, \cdots, 20\}$ and only observe $\pm 0.3\%$ performance variation. We set $N = 8$ for obtaining best performance.

5. Experimental results

Parameter Settings In Sec. 3.2, we set number of spatial grids $K = 1200$. In Sec. 4, we over-segment each frame into about 2000 superpixels via SLIC [1] for well boundary adherence. For each superpixel, we set the number of NNs $N = 8$. In our experiments, all the parameters of our algorithm are fixed to unity.

Datasets We evaluate our method on two public video segmentation benchmarks, namely DAVIS [28], and Segtrack-V2 [21]. The new released DAVIS [28] contains 50 video sequences (3,455 frames in total) and pixel-level manual ground-truth for the foreground object in every frame. Those videos span a wide range of object segmentation challenges such as occlusions, fast-motion and appearance changes. Since DAVIS contains diverse scenarios which break classical assumptions, as demonstrated in [28], most state-of-the-art methods fail to produce reasonable segments. Segtrack-V2 [21] consists of 14 videos with 24 instance objects and 947 frames. Pixel-level mask is offered for every frame.

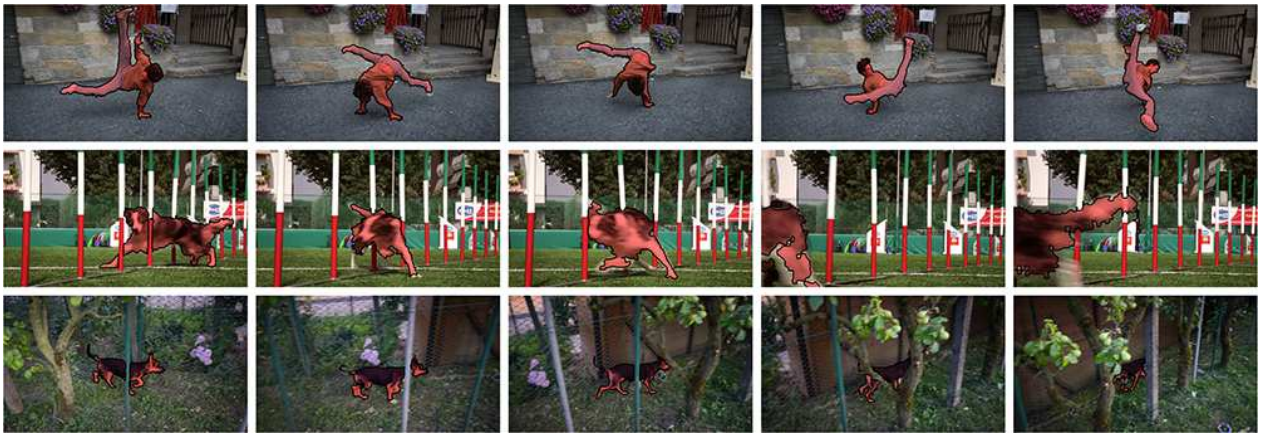


Figure 5. Qualitative segmentation results on three video sequences from DAVIS [28] (from top to bottom: *breakdance-flare*, *dog-agility* and *libby*). It can be observed that the proposed algorithm is applicable to a quite general set of sequences and robust to many challenges.

Video	IoU score						
	BVS	FCP	JMP	SEA	TSP	HVS	**STV**
breakdance-flare	0.727	0.723	0.430	0.131	0.040	0.499	**0.835**
camel	0.669	0.734	0.640	0.649	0.654	**0.876**	0.798
car-roundabout	0.851	0.717	0.726	0.708	0.614	0.777	**0.904**
dance-twirl	0.492	0.471	0.444	0.117	0.099	0.318	**0.640**
drift-chicane	0.033	0.457	0.243	0.119	0.018	0.331	**0.466**
horsejump-low	0.601	0.607	0.663	0.498	0.291	0.551	**0.768**
libby	**0.776**	0.316	0.295	0.226	0.070	0.553	0.723
mallard-fly	0.606	0.541	0.536	0.557	0.200	0.436	**0.650**
motorbike	0.563	0.713	0.506	0.451	0.340	0.687	**0.749**
rhino	0.782	0.794	0.716	0.736	0.694	0.812	**0.893**
soapbox	**0.789**	0.449	0.759	0.783	0.247	0.684	0.751
stroller	0.767	0.597	0.656	0.464	0.369	0.662	**0.826**
surf	0.492	0.843	**0.941**	0.821	0.814	0.759	0.917
swing	**0.784**	0.648	0.115	0.511	0.098	0.104	0.765
tennis	0.737	0.623	0.765	0.482	0.074	0.576	**0.826**
Avg. (entire)	0.665	0.631	0.607	0.556	0.358	0.596	**0.736**

Table 1. IoU score on a representative subset of the DAVIS dataset [28], and the average computed over all 50 video sequences. The best results are **boldfaced**.

5.1. Performance Comparison

Quantitative Results Standard Intersection-over-Union (IoU) metric is employed for quantitative evaluation. Given a segmentation mask M and ground-truth G, IoU is computed via $\frac{M \cap G}{M \cup G}$. We compare the proposed STV against various state-of-the-art alternatives: BVS [23], FCP [29], JMP [11], SEA [30], TSP [8], HVS [16], JOT [47], and OFL [36].

In Table 1, we report IoU score on a *representative* subset of the DAVIS dataset. As shown, the proposed STV performs superior on most video sequences. And STV achieves the highest average IoU score (**0.736**) over all the 50 video sequence of the DAVIS dataset, which demonstrates significant improvement over previous methods.

Method	BVS	OFL	SEA	FCP	HVS	JOT	**STV**
IoU	0.584	0.675	0.453	0.574	0.518	0.718	**0.781**

Table 2. Average IoU score for SegtrackV2 dataset. The best results are **boldfaced**.

We further report quantitative results on Segtrack-V2 [21] dataset in Table 2. The results consistently demonstrate the favorable performance of the proposed method.

Qualitative Results Qualitative video segmentation results for video sequences from the DAVIS dataset [28] and SegTrack-V2 [21] are presented in Fig. 5 and Fig. 6. With the first frame as initialization, the proposed algorithm has the ability to segment the objects with fast motion patterns (*breakdance-flare* and *cheetah1*) or large shape deformation (*dog-agility*). It also produces accurate segmentation maps even when the foreground suffers occlusions (*libby*).

5.2. Validation of the Proposed Algorithm

In this section, we offer more detailed exploration for the proposed approach in several aspects with DAVIS dataset [28]. We test the values of important parameters, verify basic assumptions of the proposed algorithm, evaluate the contributions from each part of our approach, and perform runtime comparison.

Parameter Verification We study the influence of the needed input parameter: number of spatial grids K, of our super-trajectory algorithm in Sec. 3.2. We report the performance by plotting the IoU value of the segmentation results as functions of a variety of Ks, where we vary $K = \{800, 900, \cdots, 1500\}$. As shown in Fig. 7 (a), the performance increases with finer super-trajectory clustering in spatial domain ($K\uparrow$). However, when we further increase K, the final performance does not change obviously. We set $K = 1200$ where the maximum performance is obtained. Later, we investigate the influence of parameter N, which indicates the number of the NNs of a region in Sec. 4. We

Figure 6. Qualitative segmentation results on representative video sequences from SegTrack-V2 [21] (from top to bottom: *cheetah1*, *drift1*, and *girl*). The initial masks are presented in the first row.

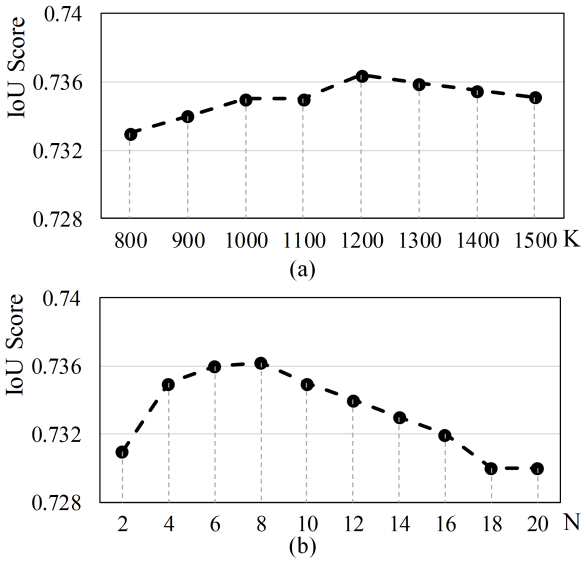


Figure 7. Parameter selection for number of spatial grids K (a) and the number of the number of the NNs N (b). The IoU score is plotted as a function of a variety of Ks (Ns).

plot IoU score with varying $N = \{2, 4, \cdots, 20\}$ in Fig. 7 (b), and set $N = 8$ for achieving best performance.

Ablation Study To quantify the improvement obtained with our proposed trajectories in Sec. 3.1, we compare to two baseline trajectories: LTM [14] and DAD [39] in our experimental results. LTM is widely used for motion segmentation and DAD shows promising performance for action detection. To be fair, we only replace our trajectory generation part with above two methods, estimate optical flow via LDOF [6] and keep all other parameters fixed. From the comparison results in Table. 3. we can find that, compared with classical trajectory methods [14, 39], the proposed trajectory generation approach is preferable.

Method	LTM	DAD	**STV**
IoU	0.718	0.654	**0.736**

Table 3. Average IoU score for DAVIS dataset with comparison to two trajectory methods: LTM [14] and DAD [39]. The best results are **boldfaced**.

6. Conclusions

This paper introduced a video segmentation approach by representing video as super-trajectories. Based on DPC algorithm, compact trajectories are efficiently grouped into super-trajectories. Occlusion and drift are naturally handled by our trajectory generation method based on a probabilistic model. We proposed to perform video segmentation on super-trajectory level. Via reverse tracking points and leveraging the property of region re-occurrence, the algorithm is robust for many segmentation challenges. Experimental results on famous video segmentation datasets [28, 21] demonstrate that our approach outperforms current state-of-the-art methods.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE PAMI*, 2012.

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

[3] X. Bai, J. Wang, D. Simons, and G. Sapiro. Video SnapCut: robust video object cutout using localized classifiers. *ACM Trans. on Graphics*, 2009.

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

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

[6] T. Brox and J. Malik. Large displacement optical flow: Descriptor matching in variational motion estimation. *IEEE PAMI*, 2011.

[7] I. Budvytis, V. Badrinarayanan, and R. Cipolla. Semi-supervised video segmentation using tree structured graphical models. In *CVPR*, 2011.

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

[9] L. Chen, J. Shen, W. Wang, and B. Ni. Video object segmentation via dense trajectories. *IEEE TMM*, 2015.

[10] A. Faktor and M. Irani. Video segmentation by non-local consensus voting. In *BMVC*, 2014.

[11] Q. Fan, F. Zhong, D. Lischinski, D. Cohen-Or, and B. Chen. JumpCut: Non-successive mask transfer and interpolation for video cutout. *ACM Trans. on Graphics*, 2015.

[12] K. Fragkiadaki, P. Arbelaez, P. Felsen, and J. Malik. Learning to segment moving objects in videos. In *CVPR*, 2015.

[13] K. Fragkiadaki and J. Shi. Detection free tracking: Exploiting motion and topology for segmenting and tracking under entanglement. In *CVPR*, 2011.

[14] K. Fragkiadaki, G. Zhang, and J. Shi. Video segmentation by tracing discontinuities in a trajectory embedding. In *CVPR*, 2012.

[15] K. Fragkiadaki, W. Zhang, G. Zhang, and J. Shi. Two-granularity tracking: Mediating trajectory and detection graphs for tracking under occlusions. In *ECCV*, 2012.

[16] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *CVPR*, 2010.

[17] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, 2015.

[18] M. Keuper, B. Andres, and T. Brox. Motion trajectory segmentation via minimum cost multicuts. In *ICCV*, 2015.

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

[20] J. Lezama, K. Alahari, J. Sivic, and I. Laptev. Track to the future: Spatio-temporal video segmentation with long-range motion cues. In *CVPR*, 2011.

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

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

[23] N. Maerki, F. Perazzi, O. Wang, and A. Sorkine-Hornung. Bilateral space video segmentation. In *CVPR*, 2016.

[24] P. Ochs and T. Brox. Higher order motion models and spectral clustering. In *CVPR*, 2012.

[25] P. Ochs, J. Malik, and T. Brox. Segmentation of moving objects by long term video analysis. *IEEE PAMI*, 2014.

[26] D. Oneata, J. Revaud, J. Verbeek, and C. Schmid. Spatio-temporal object detection proposals. In *ECCV*, 2014.

[27] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *ICCV*, 2013.

[28] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. V. Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016.

[29] F. Perazzi, O. Wang, M. Gross, and A. Sorkinehornung. Fully connected object proposals for video segmentation. In *ICCV*, 2015.

[30] S. A. Ramakanth and R. V. Babu. SeamSeg: Video object segmentation using patch seams. In *CVPR*, 2014.

[31] A. Rodriguez and A. Laio. Clustering by fast search and find of density peaks. *Science*, 2014.

[32] N. Shankar Nagaraja, F. R. Schmidt, and T. Brox. Video segmentation with just a few strokes. In *ICCV*, 2015.

[33] J. Shi and C. Tomasi. Good features to track. In *CVPR*, 1994.

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

[35] D. Tsai, M. Flagg, and J. M. Rehg. Motion coherent tracking using multi-label MRF optimization. *BMVC*, 2010.

[36] Y.-H. Tsai, M.-H. Yang, and M. J. Black. Video segmentation via object flow. In *CVPR*, 2016.

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

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

[39] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *CVPR*, 2011.

[40] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.

[41] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *CVPR*, 2015.

[42] W. Wang, J. Shen, X. Li, and F. Porikli. Robust video object cosegmentation. *IEEE TIP*, 2015.

[43] W. Wang, J. Shen, and F. Porikli. Saliency-aware geodesic video object segmentation. In *CVPR*, 2015.

[44] W. Wang, J. Shen, and L. Shao. Consistent video saliency using local gradient flow optimization and global refinement. *IEEE TIP*, 2015.

[45] W. Wang, J. Shen, L. Shao, and F. Porikli. Correspondence driven saliency transfer. *IEEE TIP*.

[46] W. Wang, J. Shen, R. Yang, and F. Porikli. Saliency-aware video object segmentation. *IEEE PAMI*, 2017.

[47] L. Wen, D. Du, Z. Lei, S. Z. Li, and M.-H. Yang. JOTS: Joint online tracking and segmentation. In *CVPR*, 2015.

[48] F. Xiao and Y. Jae Lee. Track and segment: An iterative unsupervised approach for video object proposals. In *CVPR*, 2016.

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

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

[51] F. Zhong, X. Qin, Q. Peng, and X. Meng. Discontinuity-aware video object cutout. *ACM Trans. on Graphics*, 2012.