JOTS: Joint Online Tracking and Segmentation

Longyin Wen[1], Dawei Du[2], Zhen Lei[1]*, Stan Z. Li[1], Ming-Hsuan Yang[3]
[1] NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, CHN.
[2] SCCE, University of Chinese Academy of Sciences, Beijing, CHN.
[3] School of Engineering, University of California at Merced, USA.
[1]{lywen,zlei,szli}@nlpr.ia.ac.cn, [2]dawei.du@vipl.ict.ac.cn, [3]mhyang@ucmerced.edu

Abstract

We present a novel Joint Online Tracking and Segmentation (JOTS) algorithm which integrates the multi-part tracking and segmentation into a unified energy optimization framework to handle the video segmentation task. The multi-part segmentation is posed as a pixel-level label assignment task with regularization according to the estimated part models, and tracking is formulated as estimating the part models based on the pixel labels, which in turn is used to refine the model. The multi-part tracking and segmentation are carried out iteratively to minimize the proposed objective function by a RANSAC-style approach. Extensive experiments on the SegTrack and SegTrack v2 databases demonstrate that the proposed algorithm performs favorably against the state-of-the-art methods.

1. Introduction

Recent advances in video segmentation aim to extract target objects from the background with accurate boundaries using mainly offline approaches [24, 13, 21, 16, 26, 19, 27, 20]. Despite much demonstrated success, existing methods are less effective for applications that entail online processing. Examples abound, including video surveillance, action recognition and human-computer interaction, to name a few. Recently, some online video segmentation methods are proposed, e.g., [7] and SPT [18]. The global object appearance modeling without strong local constraints in [7] and the target-independent proposals generation step in SPT [18] may cause inaccurate segmentation results, especially in the scene with complex background or large motions.

Image segmentation aims to partition pixels based on certain characteristics (e.g., color, intensity, or texture) in the spatial domain, while tracking intends to partition pixels based on the consistence properties in the temporal domain. Clearly each task facilitates the other especially for online video segmentation, and both modules should be considered in the same framework. Thus, in this work, we propose a Joint Online Tracking and Segmentation (JOTS) algorithm, which formulates the video segmentation task as the online multi-part tracking and segmentation in a unified energy function. The online multi-part tracking provides effective sequential motion and structure constraints for segmentation, while the multi-part segmentation generates accurate local appearance and location information to facilitate tracking. The tracking and segmentation stages are optimized iteratively by a RANSAC-style approach to validate the results generated by each module for accurate performance. The main steps of the proposed algorithm are shown in Figure 1. As an example in Figure 1(c), the target multi-part model learned in the first two iterations fail to fit the current pixels well (center locations of some models are not located at the centroids of the pixels with the same label). As the multi-part models fit pixels better through tracking, the labeling error decreases. Meanwhile, more accurate multi-part tracking is achieved as the labeling error decreases.

The contributions of this work are summarized as follows. First, a novel joint online tracking and segmentation algorithm is proposed for online video segmentation, where the multi-part tracking and segmentation are integrated in a unified energy objective function to achieve better performance. Second, the minimization of the proposed energy function is effectively solved by a RANSAC-style approach with the α-expansion algorithm. Third, extensive experiments on two benchmark datasets, i.e., SegTrack and SegTrack v2, against the state-of-the-art methods are carried out to demonstrate the effectiveness of our method.

2. Related Work and Problem Context

Segmentation. Video segmentation has attracted much attention due to its importance in vision problems. Numerous algorithms have been proposed to address this problem using both past and future frames of an image sequence with batch processing [24, 13, 21, 16, 26, 19, 27, 20]. While

*Corresponding author.

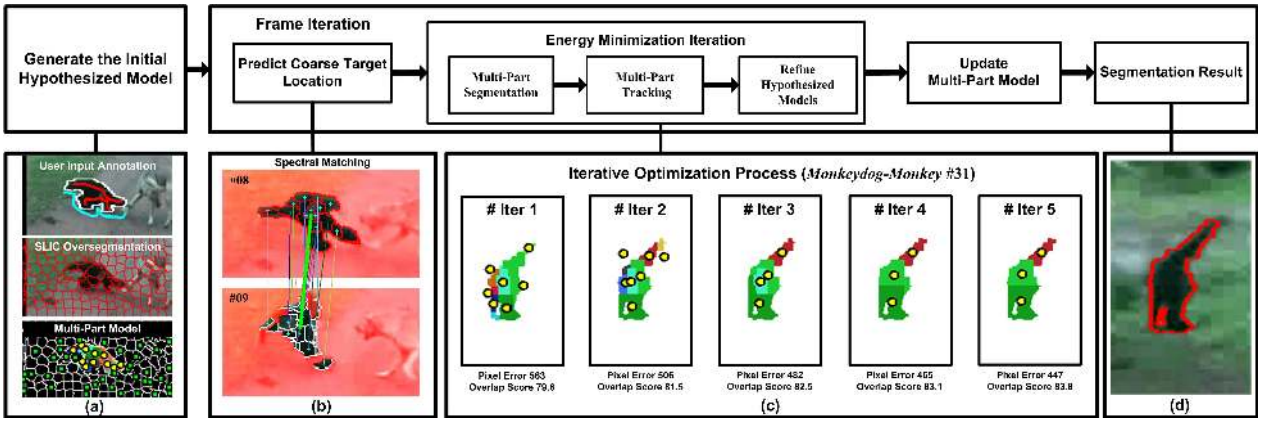

Figure 1. Main steps (upper part) and corresponding example (lower part) of the proposed JOTS algorithm. (a) Initial multi-part model construction. The yellow and green circles represent the center locations of the target and background parts respectively. (b) Spectral matching for generating approximate target location. The green line represents the correspondence between the predicted coarse target location and the center location in the previous frame, and the other lines denote some matches of target parts. (c) An example showing the iterative optimization process. The parts of a target object are denoted with different colors. (d) Final segmentation result where the target boundary is delineated by red pixels.

these methods generate promising results, they are not applicable to online vision tasks such as video surveillance, action recognition, and human-computer interaction.

In [7], an integrated probabilistic model for online video segmentation is proposed, which combines dynamics of implicit shapes, topological shape constraints, adaptive appearance model, and layered flow. As only pixel-level information without local constraints is used to distinguish between the foreground and background classes, it is likely to include false positives especially for videos with cluttered backgrounds. Li *et al*. [18] propose an unsupervised video segmentation approach by associating a pool of object proposals in consecutive frames. However, target-independent proposals may render inaccurate segmentation results.

Tracking. Several tracking-by-segmentation methods [9, 12, 11, 28] have been developed in the literature. In [9], a level-set formulation is presented to accurately extract object boundaries for tracking. Godec *et al*. [12] extend the Hough forest classifier to the online setting and integrate voting-based detection and GrabCut segmentation [23] methods for tracking. In [11], Duffner and Garcia propose a fast adaptive method based on Hough transform with pixel-based descriptors and segmentations (similar to [2]) to handle non-rigid deformation for object tracking. Zhong *et al*. [28] integrate segmentation with tracking to alleviate the drifting problem based on structured labeling information using partial least squares regression analysis. The aforementioned methods focus on using segmentation techniques to help tracking rather than extracting targets from the background accurately.

In addition to tracking-by-segmentation methods, part-based tracking approaches [22, 25, 6, 15] have been proposed in recent years. However, these methods mainly generate regions based on certain image properties, rather than focus on objects of interest and thus the boundaries are less accurate (i.e., some regions contain both foreground and background pixels). One approach is to segment each frame into superpixels and use a conditional random field algorithm to separate foreground and background regions [22]. The tracking process is based on matching of color distributions for both classes. Wang *et al*. [25] propose a discriminative appearance model based on superpixels in which the probabilities of superpixels belonging to the foreground are used to separate the target from the background. In [6], Cai *et al*. design a dynamic graph based method to account for non-rigid motion. Hong *et al*. [15] present a hierarchical appearance representation model for tracking based on a graphical model that exploits shared information across multiple quantization levels including pixels, superpixels, and bounding boxes. The above-mentioned methods focus on tracking non-rigid moving targets with the help of segmentation. As coarse mid-level segmentation based on superpixels is used rather than fine details from low-level pixels, the generated object boundaries are less accurate.

Recently, a tracking method based on temporal correlation of superpixels [8] is developed. In contrast, the proposed JOTS algorithm focuses on accurately segmenting target objects from the background on the pixel level.

3. Problem Formulation

Given simple user annotation followed by the interactive segmentation method [14] in the first frame, we first segment a target object from the background, and then use the SLIC algorithm [1] to generate the initial hypothesized models, as depicted in Figure 1(a) (see also Section 5.3).

Let $\boldsymbol{\mathcal{M}} = \{\mathcal{M}_1, \ldots, \mathcal{M}_k\}$ be the k parts of a target object with the label set $\{l_1, \ldots, l_k\}$, where $\mathcal{M}_i = (\mathcal{A}_i, \mathcal{P}_i, \Theta_i)$ is the i-th model, $\mathcal{A}_i$ is the HSV histogram of the model, $\mathcal{P}_i$ is the center location of the model, and Θ_i is the location set of the pixels belonging to the model. We also construct a model $\mathcal{M}' = \{\mathcal{M}'_1, \ldots, \mathcal{M}'_n\}$ to describe the pixels from the complex background with label l_0 as outliers, where n is the number of parts in the background model, and $\mathcal{M}'_i = (\mathcal{A}'_i, \mathcal{P}'_i, \Theta'_i)$ is the i-th part, $\mathcal{A}'_i$ is the corresponding HSV histogram, $\mathcal{P}'_i$ is the respective center location, and Θ'_i is the location set of the pixels belonging to the model.

Different from previous works, which segment a target object based on the given target model (e.g., appearance or location models), the video segmentation task in this work is formulated by multi-part tracking and segmentation in a unified framework. That is, we optimize the pixel labels f and the target multi-part model $\boldsymbol{\mathcal{M}}$ simultaneously. For each pixel p in the image, we assign it with a label $f_p \in \{l_0\} \cup \{l_1, \ldots, l_k\}$ indicating which part it belongs to (i.e., multi-part segmentation) rather than merely identify it as the foreground or background in previous methods, and optimize the target multi-part model $\boldsymbol{\mathcal{M}}$ in the current frame (i.e., multi-part tracking) simultaneously. After obtaining the label assignment of each pixel, the video segmentation task for each frame is naturally completed.

To reduce the computational complexity, we only assign the labels to the pixels near the predicted target location at the current frame, which is determined by the previous multi-part model $\overline{\boldsymbol{\mathcal{M}}}$. The video segmentation problem is formulated as follows:

$$\{\boldsymbol{\mathcal{M}}^*, f^*\} = \underset{\boldsymbol{\mathcal{M}}, f}{\operatorname{argmin}} E(\boldsymbol{\mathcal{M}}, f|\overline{\boldsymbol{\mathcal{M}}}), \tag{1}$$

where $E(\boldsymbol{\mathcal{M}}, f|\overline{\boldsymbol{\mathcal{M}}})$ is obtained from the segmentation stage that integrates both multi-part tracking and segmentation into a unified energy function. $\boldsymbol{\mathcal{M}}^*$ and f^* are the multi-part model and the pixel labeling result in the current frame, respectively. To solve (1), we first obtain the predicted target location using the dynamic structure graph matching method [6]. Specifically, we first use the SLIC algorithm [1] to generate multiple candidate parts in the current frame, and use the spectral matching technique [17] to find the matches between the parts in the previous target model and the candidate parts, as shown in Figure 1(b). The coarsely estimated target location is computed based on the votes of the matched parts. Finally, we set the bounding box centered at the coarsely estimated target location with η larger (empirically set as 1.8) than the target size in the previous frame as the segmentation region.

In the segmentation region, the optimal labels f^* for each pixel and the optimal multi-part model $\boldsymbol{\mathcal{M}}^*$ in the current frame are obtained by minimization the energy $E(\boldsymbol{\mathcal{M}}, f|\overline{\boldsymbol{\mathcal{M}}})$. A RANSAC-style method is proposed to obtain the solution in two steps: 1. the pixel labels are assigned with the current estimated multi-part model by the α-expansion algorithm [4]; 2. the target parts are tracked according to the pixel appearance likelihood and motion consistency with the current labeling. These two steps are iterated until reaching the minimal energy of the objective function such that the multi-part tracking facilitates the multi-part segmentation, and vice versa. After the iterative optimization, we update the multi-part model based on the optimal labeling and output the final segmentation result (See Figure 1(d)). The proposed optimization process is described in the following section.

4. Joint Online Tracking and Segmentation

We compute the solution $\{\boldsymbol{\mathcal{M}}^*, f^*\}$ by minimizing $E(\boldsymbol{\mathcal{M}}, f|\overline{\boldsymbol{\mathcal{M}}})$ in (1). For the clarity of presentation, we omit $\overline{\boldsymbol{\mathcal{M}}}$ in the following equations. The objective energy $E(\boldsymbol{\mathcal{M}}, f)$ includes both multi-part tracking and segmentation information with a regularization term, that is

$$\begin{aligned}\{\boldsymbol{\mathcal{M}}^*, f^*\} = \underset{\boldsymbol{\mathcal{M}}, f}{\operatorname{argmin}} E(\boldsymbol{\mathcal{M}}, f) = \underset{\boldsymbol{\mathcal{M}}, f}{\operatorname{argmin}} \Big\{ D(f, \boldsymbol{\mathcal{M}}) \\ + \sum_{(p,q)\in\mathcal{N}} V_{p,q}(f_p, f_q) + H(f, \boldsymbol{\mathcal{M}}) \Big\},\end{aligned} \tag{2}$$

where $D(f, \boldsymbol{\mathcal{M}})$ is the data term based on the current labeling f and multi-part model $\boldsymbol{\mathcal{M}}$, $V_{p,q}(f_p, f_q)$ is the smooth term describing the interactions between neighboring pixels, and $H(f, \boldsymbol{\mathcal{M}})$ is the regularization term of $D(f, \boldsymbol{\mathcal{M}})$ to avoid overfitting [3, 10] by enforcing constraints of the models in pixel labeling.

4.1. Data Term

The spatio-temporal continuity of two aspects, i.e., target appearance and location, provides effective information for the online video segmentation task. If the pixel p is labeled as l_i, we expect that the pixel has small energy of belonging to the part model $\mathcal{M}_i$ in both aspects. The data term in (2) is defined as

$$\begin{aligned}D(f, \boldsymbol{\mathcal{M}}) &= \sum_{p\in\mathcal{S}} D_p(l_i, \mathcal{M}_i) \\ &= \sum_{p\in\mathcal{S}} \big(\alpha_1 \cdot \Phi_a(\rho_p; \mathcal{A}_i) + \alpha_2 \cdot \Phi_l(\ell_p; \mathcal{P}_i)\big),\end{aligned} \tag{3}$$

where $D_p(l_i, \mathcal{M}_i)$ is the data energy of the pixel p labeled as l_i, $\mathcal{S}$ is the pixel set in the segmentation region, α_1, α_2 are weight parameters, $\Phi_a(\rho_p; \mathcal{A}_i)$, and $\Phi_l(\ell_p; \mathcal{P}_i)$ are the energy terms based on appearance and location, respectively. In this formulation, $\Phi_a(\rho_p; \mathcal{A}_i)$ is the appearance energy induced by the appearance likelihood of a pixel p belonging to the model $\mathcal{M}_i$, which is computed as the bin value of the pixel in the HSV histogram $\mathcal{A}_i$. In addition, $\Phi_l(\ell_p; \mathcal{P}_i)$ is the location energy induced by the location likelihood of a pixel p based on the displacement from the center location

$\mathcal{P}_i$, computed by the product of two single Gaussian models in the vertical and horizontal.

Similarly, the data energy of a pixel belonging to the background (i.e., outlier) is described by appearance as well as location, and defined by the minimal energy of all background sub-models, i.e., $D_p(f_p, \mathcal{M}') = \min_j D_p(f_p, \mathcal{M}'_j)$.

4.2. Smooth Term

Intuitively, if the two adjacent pixels have similar appearance, the same label is assigned to them with small energy. On the other hand, the motions of targets from background are usually distinct, especially at the object boundaries (i.e., motion discontinuity). These appearance and motion cues provide effective information to distinguish the target objects and background pixels. Based on these factors, the smooth term $V_{p,q}(f_p, f_q)$ in (2) is defined as

$$V_{p,q}(f_p, f_q) = \mathbb{I}(f_p \neq f_q) \cdot \Big(\alpha_3 \cdot \Delta_c(p,q) + \alpha_4 \cdot \Delta_f(p,q)\Big), \tag{4}$$

where $\mathbb{I}(\cdot)$ returns 1 if its argument is true, and 0 otherwise, and $\Delta_c(p,q)$ and $\Delta_f(p,q)$ are the Euclidean distance between the two adjacent pixels p and q in the RGB color space and optical flow field [5] respectively. In the above formulation, α_3, α_4 are the weight parameters.

4.3. Regularization Term

To avoid the overfitting problem [10, 3], we regularize the data term. The regularization term in (2) consists of three factors: 1. *Area*: it encourages all the used models with the similar size; 2. *Profile*: it penalizes the incomplete and irregular models in pixel labeling; 3. *Complexity*: it penalizes the number of models used in pixel labeling.

$$H(f, \boldsymbol{\mathcal{M}}) = \sum_{i=1}^{k} \mathbb{I}\big(\exists p : f_p = l_i\big) \cdot \Big(\alpha_5 \cdot H_a(f, \mathcal{M}_i) + \alpha_6 \cdot H_p(f, \mathcal{M}_i) + \alpha_7 \cdot H_c(f, \mathcal{M}_i)\Big), \tag{5}$$

where $\mathbb{I}(\cdot)$ returns 1 if its argument is true, and 0 otherwise, $H_a(f, \mathcal{M}_i)$, $H_p(f, \mathcal{M}_i)$ and $H_c(f, \mathcal{M}_i)$ are the area, profile, and complexity regularization terms of the part model $\mathcal{M}_i$ respectively. In this formulation, α_5, α_6, and α_7 are the corresponding weights. For all the background sub-models, we set $H(f, \mathcal{M}'_i) = 0$, where $i = 1, \cdots, n$. These regularization terms are described as follows.

Area. The model with large area does not handle large object deformation well. On the other hand, the model with small area is susceptive to background noise. Hence, we define the area regularization term as

$$H_a(f, \mathcal{M}_i) = \Big| |\Theta_i| - \frac{1}{k} \sum_{j=1}^{k} |\Theta_j| \Big|, \tag{6}$$

where $|\Theta_j|$ represents the number of pixels in $\mathcal{M}_j$.

Profile. As some target parts may be occluded when large deformation occurs, the object extent and center location may not be estimated accurately. To encourage the new models and suppress the inaccurate ones, we define the profile regularization term as

$$H_p(f, \mathcal{M}_i) = \Delta_p(\mathcal{P}_i, \mathcal{C}_i) \cdot \Omega\Big(\forall p \in B_i, \Delta_p(\ell_p, \mathcal{C}_i)\Big), \tag{7}$$

where $\mathcal{P}_i$ and $\mathcal{C}_i$ are the center location of the part model $\mathcal{M}_i$ and the region constructed by the pixels labeled as l_i, respectively. $\Delta_p(\cdot,\cdot)$ is the Euclidean distance function to calculate the distance between two points in the 2D image plane. In addition, ℓ_p is the location of pixel p, and B_i is the boundary pixel set of the region constructed by the pixels labeled as l_i. $\Omega(\cdot)$ is the variance function to calculate the distance variance of the boundary pixels.

Complexity. The constant complexity regularization term is used to penalize labeling results with large number of target models, i.e., $H_c(f, \mathcal{M}_i) = 1$.

5. Energy Minimization

The energy minimization problem in (2) is challenging as the objective function involves two sets of variables. We optimize f and $\boldsymbol{\mathcal{M}}$ alternatively in spirit similar to [10, 3] with multi-part tracking and multi-part segmentation.

An initial multi-part model $\boldsymbol{\mathcal{M}}^{[0]}$ is obtained from the optimal models in the previous frame. Clearly incorrect models may be contained in $\boldsymbol{\mathcal{M}}^{[0]}$. In the multi-part segmentation stage, pixel labels $f^{[0]}$ are computed by the α-expansion algorithm with the regularization term and a small set of reliable models in $\boldsymbol{\mathcal{M}}^{[0]}$ are selected. In the multi-part tracking stage, the selected models are improved by re-estimating the HSV histogram and location models with the energy function (9). Next, we add some hypothesized part models based on the current labeling to expand the multi-part model $\boldsymbol{\mathcal{M}}^{[1]}$. These two stages are repeated to generate a series of labelings $f^{[0]}, f^{[1]}, f^{[2]}, \cdots$ and multi-part model set $\boldsymbol{\mathcal{M}}^{[0]}, \boldsymbol{\mathcal{M}}^{[1]}, \boldsymbol{\mathcal{M}}^{[2]}, \cdots$ until the energy in (2) is no longer reduced (See Figure 1(c)). Thus the energy is minimized to obtain the optimal labeling f^* and multi-part model $\boldsymbol{\mathcal{M}}^*$.

The energy function $E(\boldsymbol{\mathcal{M}}, f)$ is non-negative with a natural lower bound of 0. Meanwhile, the energy is non-increasing over the iterations to ensure the convergence of this optimization process. We present some examples in Figure 2 to show how the energy is iteratively reduced. Meanwhile, as the overall energy is decreased, the corresponding intersection-over-union overlap score of the JOTS algorithm increases, which demonstrates the effectiveness of our energy minimization method. The multi-part tracking and segmentation modules are described next.

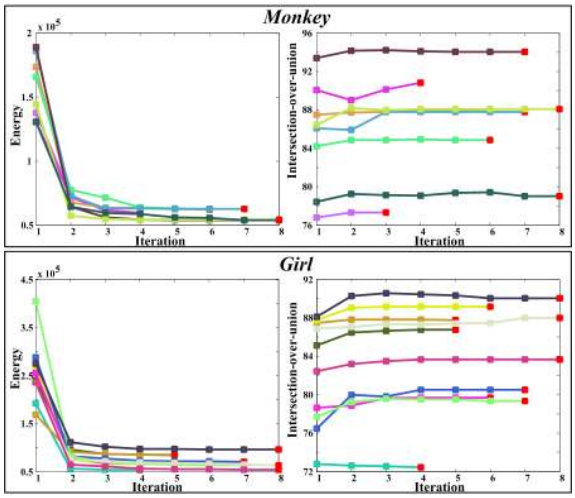

Figure 2. Left to right: energy curves and segmentation results based on the intersection-over-union overlap metric in the iterative optimization process of several sample frames in the *Monkey* and *Girl* sequences. Different frames are described by different colors. The red square on each curve represents the end of each iterative process.

5.1. Multi-Part Segmentation

To segment multiple target parts from the background, we assign a pixel p in the segmentation region with multiple labels $\{l_0, l_1, \ldots, l_k\}$ rather than simply classify it as the foreground or background in previous methods. The pixel labeling problem is formulated as an energy minimization of the pairwise Markov random field,

$$f^* = \underset{f}{\operatorname{argmin}}\, D(f, \boldsymbol{\mathcal{M}}) + \sum_{(p,q)\in\mathcal{N}} V_{p,q}(f_p, f_q) + H(f, \boldsymbol{\mathcal{M}}), \tag{8}$$

where $D(f, \boldsymbol{\mathcal{M}})$ is the data term based on the labeling f and multi-part model $\boldsymbol{\mathcal{M}}$, $V_{p,q}(f_p, f_q)$ is the smooth term describing interactions between neighboring pixels, $\mathcal{N}$ is the 4-neighborhood relations between pixels in $\mathcal{S}$. The optimization problem can be solved by the α-expansion algorithm [4] with graph cut effectively since the energy function remains sub-modular.

5.2. Multi-Part Tracking

Once the pixel labeling f in the segmentation region are computed, we re-estimate the multi-part model $\boldsymbol{\mathcal{M}} = \{\mathcal{M}_1, \ldots, \mathcal{M}_k\}$ by minimizing the energy $E(\boldsymbol{\mathcal{M}}, f)$. Given the current pixel labels f, the smooth term in (2) is fixed and the multi-part tracking problem is formulated as

$$\boldsymbol{\mathcal{M}}^* = \underset{\boldsymbol{\mathcal{M}}}{\operatorname{argmin}}\, D(f, \boldsymbol{\mathcal{M}}) + H(f, \boldsymbol{\mathcal{M}}), \tag{9}$$

It is challenging to solve (9) since the regularization term is difficult to minimize with respect to the multi-part model $\boldsymbol{\mathcal{M}}$. Similar to [3], we disregard the regularization term at first and focus on minimizing the first term of (9) using the Maximum Likelihood Estimation (MLE) method to obtain the optimal models $\boldsymbol{\mathcal{M}}^*$. That is, for the i-th selected model $\mathcal{M}_i$ in labeling f, we estimate the HSV histogram $\widehat{\mathcal{A}}_i$, the center location $\widehat{\mathcal{P}}_i$, and the pixel location set $\widehat{\Theta}_i$ based on the current observed pixels labeled as l_i. Then, if the overall energy of (2) is reduced, we use the estimated model $(\widehat{\mathcal{A}}_i, \widehat{\mathcal{P}}_i, \widehat{\Theta}_i)$ to replace $\mathcal{M}_i$; otherwise, the part model $\mathcal{M}_i$ is retained.

This optimization scheme is motivated by two factors: 1. the simplified minimization (dropping the regularization term) is effectively solved by the maximum likelihood estimation method while keeping the overall energy non-increasing; 2. the simplification of the minimization should have insignificant effect on the complete energy minimization scheme. That is, if the current solution is near the good minima, the gradient of the regularization term $\frac{\partial}{\partial \boldsymbol{\mathcal{M}}} H(f, \boldsymbol{\mathcal{M}})$ will be small since the solution already obeys the constraints discussed in Section 4.3. Otherwise, a large gradient $\frac{\partial}{\partial \boldsymbol{\mathcal{M}}} H(f, \boldsymbol{\mathcal{M}})$ indicates that there exists another model which is more plausible to the constraints. The model will be picked up by the following step to refine hypothesized models (described in next section). Thus, we defer the difficult aspects of energy minimization to the subsequent multi-part segmentation optimization stage.

5.3. Expanding Hypothesized Part Model

Generating the initial hypothesized model. In the first frame, multiple parts are generated in the initial target area by the SLIC algorithm [1]. If the overlap ratio between a generated part and the user annotated target region is larger than a threshold θ_1 (e.g., 0.5 in this work), we add it to generate the initial part models and otherwise consider it as part of the background.

Refining hypothesized models. To obtain a better part model from existing ones for segmentation, we use two criteria to merge and split regions: 1. Only the neighboring small regions with similar appearance are randomly chosen to generate new models. A region is considered small if the number of pixels is less than the average number of pixels of the current multi-part model. 2. A labeled region with area larger than twice of the average area of the current used part models is split into multiple ones by the SLIC algorithm.

6. Experiments

We evaluate the proposed algorithm on two video segmentation benchmark databases, namely the SegTrack [24] and SegTrack v2 [18] databases. As discussed in [18], the pixel errors on objects of different size vary considerably. In addition, the pixel error metric is sensitive to the manually annotation errors. For fair and comprehensive comparisons, the results on the average pixel error per frame are reported in the original SegTrack database and the results

Figure 3. Segmentation results of the JOTS algorithm in 3 sequences from the SegTrack and SegTrack v2 databases. The estimated parts of each frame are presented. Different part models are described by different colors.

on the intersection-over-union overlap metric are reported on the SegTrack v2 database.

Quantitative evaluations against several state-of-the-art methods [16, 13, 27, 18, 24, 12, 7, 25, 6] are presented in Table 1 and Table 2. The top two performing methods are shown in red and blue, respectively. Some segmentation results are presented in Figure 3.

6.1. Implementation Details

In each experiment, the initial hypothesized model is generated with the simple user annotation followed by the segmentation method [14] and the SLIC algorithm [1] for our method (See Figure 1(a)). All experiments are carried out on a machine with a 2.9 GHz Intel i7 processor and 16 GB memory. The run time complexity of the proposed JOTS algorithm depends on the size of the target object. For example, implemented in MATLAB without code optimization, it takes about 20 seconds per frame to segment the target in the *Monkeydog-Monkey* sequence containing images of 320×240 pixels. The source code of the proposed JOTS algorithm will be made publicly available.

For each sequence, the number of superpixels for the SLIC algorithm [1] in initialization is set according to the size of the target. Empirically, the JOTS algorithm performs well when each part model for both target and background contains about 50 to 200 pixels. All the other parameters in the JOTS algorithm are fixed in all experiments. We use 6 bins for each channel of the HSV histogram to describe a target object. For the preset weight parameters, we take the following default values: $\alpha_1 = 2.0$, $\alpha_2 = 1.2$; $\alpha_3 = 3.0$, $\alpha_4 = 3.0$; $\alpha_5 = 20$, $\alpha_6 = 20$, $\alpha_7 = 10$. It takes 2 to 8 iterations to solve (8) (See also Figure 2).

6.2. Databases

SegTrack database. The SegTrack database [24] consists of 6 challenging videos (*Birdfall*, *Cheetah*, *Girl*, *Monkeydog*, *Parachute*, and *Penguin*) with pixel-level human annotated segmentation results for the primary foreground objects. It includes multiple interacting objects (*Cheetah* and *Penguin*), abrupt motion (*Monkeydog* and *Birdfall*), complex deformation (*Girl* and *Monkeydog*), and appearance change (*Parachute*).

SegTrack v2 database. This database is an extension of the SegTrack database with more annotated objects and 8 new video sequences are included: *Bird of Paradise*, *BMX*, *Drift*, *Hummingbird*, *Monkey*, *Frog*, *Worm*, and *Solider*. There are 14 sequences with 24 objects over 947 annotated frames in this database including different challenging factors for video segmentation, including multiple interacting objects (*Cheetah*, *Drift*, and *Penguin*), appearance change (*Bird of Paradise* and *Drift*), occlusion (*Cheetah*, *BMX*, and *Drift*), and complex deformation (*BMX-Person*, *Hummingbird*, *Frog*, *Solider*, *Monkey*, and *Worm*).

6.3. Quantitative Comparison

Table 1 shows the quantitative results of the proposed algorithm and state-of-the-art video segmentation methods [16, 27, 18, 24, 12, 7, 25, 6]. Overall, the proposed JOTS algorithm performs favorably against most online and offline methods on the SegTrack database using the pixel error metric. In addition, Table 2 shows that the JOTS algorithm performs well against the other methods [18, 16, 13, 12, 25, 6] on the SegTrack v2 database[1] using the intersection-over-union overlap metric. Detailed analysis and discussions on the quantitative evaluation are presented next.

Multiple interacting objects. The targets in the *Penguin* and *Cheetah* sequences have similar appearance to neighboring objects. For the offline methods SPT+CSI [18] and [16, 13, 27, 24], all the available object proposals in both the past and future frames are used for segmentation. Inevitably, inaccurate object proposals and wrong association in these methods easily result in larger segmentation errors, especially in the cases where the correct target is surrounded by multiple adjacent/interacting objects with similar appearance, as shown in Table 2. The same problem also exists for the online method SPT [18] that relies on the object proposals. In contrast, the JOTS algorithm performs well in these sequences. This can be attributed to that the JOTS algorithm exploits the temporal consistence of both target parts and their neighboring background, which helps distinguish these regions when they are similar.

Abrupt motion. The targets in the *Birdfall* and *Monkeydog-Monkey* sequences exhibit fast and abrupt motion. The Gaussian location model in the JOTS algorithm may not

[1] We cannot obtain the source code or binary executable to produce the results of online method [7] in the SegTrack v2 database. Thus, only the available results in the SegTrack database of [7] are reported.

Table 1. Average pixel error per frame in the SegTrack database. In the table, – indicates that the result is not reported in the sequence. The results with ∗ indicates exclusion of the *Penguin* sequence. Same as [27], the average score is the mean pixel error per frame.

Sequences	[16]	[27]	SPT+CSI [18]	[24]	[12]	SPT [18]	[7]	[25]	[6]	JOTS
Supervised	×	×	×	√	√	×	√	√	√	√
Online	×	×	×	×	√	√	√	√	√	√
Birdfall	288	**155**	242	252	466	188	265	1204	481	**163**
Cheetah	905	**633**	1156	1142	1431	983	**570**	2765	2825	806
Girl	1785	1488	1564	**1304**	6338	1573	**841**	10505	7790	1904
Monkeydog	521	365	483	563	809	558	**289**	2466	5361	**342**
Parachute	**201**	**220**	328	235	1028	339	310	2369	3105	275
Penguin	136285	-	5116	1705	6239	5026	**456**	9078	11669	**571**
Average	23949	452*	1391	785	2297	1374	**400**	4156	5282	**535**

be able to handle large displacement from the center location of the target in the previous frame. In such cases, the coarse center location of the target is predicted by the spectral matching method before segmentation (See Figure 1(b)), which enables the JOTS algorithm to handle the abrupt motion challenge. In contrast, the method [7] tracks the target only by the optical flow, which makes it fail to handle the large displacement of the target in conservative frames, and may be easily affected by the background noise in complex scenes, e.g., the object in the *Birdfall* sequence.

Complex deformation. It is challenging to segment the non-rigid objects in the *Girl*, *Monkeydog-Monkey*, *Hummingbird*, *Frog*, *Worm*, *Solider*, *Monkey* and *BMX-Person* sequences due to large deformation. Notwithstanding that these sequences contain complex object deformations or cluttered background challenges, the part-based representation in JOTS algorithm is able to handle such cases effectively. It is worth mentioning that the online superpixel-based tracking methods, e.g., [25] and [6], do not perform well in these sequences against the JOTS algorithm (See Table 1 and Table 2). The superpixels are generated independently in each frame by these methods. Thus, some of the superpixels contain both foreground and background pixels, and the segmentation results are less accurate. In addition, the HT method [12] also does not perform well mainly due to the pixel-based representation without effective local constraints in segmentation (i.e., pixels from the foreground and background are easily confused based only on the global appearance information). With the combination of pixel-level segmentation and part-level tracking in a unified iterative optimization formulation, both representations facilitate each other for accurate segmentation and tracking results in our method.

Appearance change. The large appearance change challenge happens in the *Parachute*, *Bird of Paradise* and *Drift* sequences. The proposed JOTS algorithm performs well against the other video segmentation methods [13, 18, 16] in these sequences (See Table 2). The algorithm [13] directly aggregates superpixels from both the foreground and background without considering the target object specifically, and thus the segmentation results are less accurate. In

Table 2. Intersection-over-union overlap metric of the segmentation of each algorithm in the SegTrack v2 database. In the table, – indicates that the method fails to complete the segmentation task in the sequence, and ∗ indicates exclusion of the failed sequences.

Sequence/Object	SPT+CSI [18]	[16]	[13]	SPT [18]	[12]	[25]	[6]	JOTS
Supervised	×	×	×	×	√	√	√	√
Online	×	×	×	√	√	√	√	√
Girl	**89.2**	87.7	31.9	**89.1**	53.6	52.4	62.0	84.6
Birdfall	**62.5**	49.0	57.4	62.0	56.0	32.5	36.4	**78.7**
Parachute	93.4	**96.3**	69.1	93.2	85.6	69.9	59.3	**94.4**
Cheetah-Deer	37.3	44.5	18.8	40.1	**46.1**	33.1	38.7	**66.1**
Cheetah-Cheetah	40.9	11.7	24.4	**41.3**	**47.4**	14.0	19.7	35.3
Monkeydog-Monkey	71.3	**74.3**	68.3	58.8	61.0	22.1	25.7	**82.2**
Monkeydog-Dog	**18.9**	4.9	18.8	17.4	**18.9**	10.2	3.83	**21.1**
Penguin-#1	51.5	12.6	**72.0**	51.4	54.5	20.8	40.1	**94.2**
Penguin-#2	76.5	11.3	**80.7**	73.2	67.0	20.8	37.9	**91.8**
Penguin-#3	**75.2**	11.3	**75.2**	69.6	7.59	10.3	31.2	**91.9**
Penguin-#4	57.8	7.7	**80.6**	57.6	54.3	13.0	30.2	**90.3**
Penguin-#5	**66.7**	4.2	62.7	63.4	29.6	18.9	10.7	**76.3**
Penguin-#6	50.2	8.5	**75.5**	48.6	2.09	32.3	35.0	**88.7**
Drift-#1	**74.8**	63.7	55.2	**73.8**	62.6	43.5	57.2	67.3
Drift-#2	**60.6**	30.1	27.2	58.4	21.8	11.6	13.8	**63.7**
Hummingbird-#1	**54.4**	46.3	13.7	45.4	11.8	28.8	25.1	**58.3**
Hummingbird-#2	**72.3**	**74.0**	25.2	65.2	-	45.9	44.2	50.7
Frog	**72.3**	0	**67.1**	65.8	14.5	45.2	38.8	56.3
Worm	**82.8**	**84.4**	34.7	75.6	36.8	27.4	44.3	79.3
Soldier	**83.8**	66.6	66.5	**83.0**	70.7	43.0	54.2	81.1
Monkey	**84.8**	79.0	61.9	84.1	73.1	61.7	58.7	**86.0**
Bird of Paradise	**94.0**	92.2	86.8	88.2	5.10	44.3	46.5	**93.0**
BMX-Person	85.4	**87.4**	39.2	75.1	2.04	27.9	36.0	**88.9**
BMX-Bike	24.9	**38.6**	**32.5**	24.6	-	6.04	3.86	5.70
Mean per object	**65.9**	45.3	51.8	62.7	40.1*	30.7	35.6	**71.8**
Mean per sequence	**71.2**	57.3	50.8	68.0	41.0*	37.0	40.4	**72.2**

contrast, the JOTS algorithm integrates target parts state estimation (from multi-part tracking) and pixel labeling (from multi-part segmentation) in a unified energy minimization formulation, such that more accurate segmentation results can be generated in the RANSAC-style iterative optimization process (See Figure 2). The online method SPT [18] and offline methods SPT+CSI [18] and [16] deal with the appearance changes of the target by refining and associating the object proposals in consecutive frames. In contrast, the JOTS algorithm performs well in handling appearance changes of targets online by selecting good part models in the optimization process (as discussed in Section 5.3).

Occlusion. As presented in Table 2, the proposed JOTS algorithm performs well when target objects are occluded (e.g., in the *Cheetah*, *Drift* and *Penguin* sequences), which can be explained by the RANSAC-style optimization approach for selecting a small set of reliable target models to generate more accurate results (as discussed in Section 5). However, when a target object undergoes heavy occlusion in the very first frame of the *BMX-Bike* sequence, the unsupervised methods SPT [18], SPT+CSI [18] and [16, 13] perform well against our method. Since the occluded parts are contained in the object proposals, all these methods are able to associate the unoccluded and occluded parts of the target in the *BMX-Bike* sequence to achieve better performance. However, our method considers only the spatio-temporal consistence between consecutive frames to segment objects online. Thus, it is difficult for the proposed JOTS algorithm

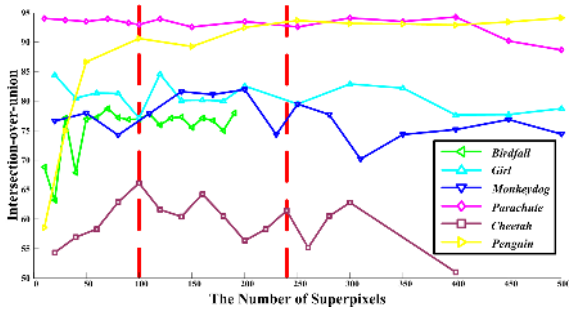

Figure 4. Sensitive analysis of the part model size in the SegTrack database based on the intersection-over-union overlap metric. The region between two red dashed lines is the suggested value range for the number of initial superpixels.

to recognize the separate occluded parts since the information of them is not available in the first frame. Overall, our method performs well in both databases when occlusion challenge happens.

6.4. Discussion

Sensitive analysis of the part model size. We study the influence of the number of initial superpixels in the proposed JOTS algorithm, which resolves the size of initial part models. As presented in Figure 4, the JOTS algorithm is relatively robust to the small perturbations of the number of initial superpixels. Specifically, we notice that the performance of the JOTS algorithm improves to reach a stable state as the number of superpixels increases for rigid target objects, e.g., the *Parachute*, *Penguin*, and *Birdfall* sequences. While for non-rigid objects, e.g., the *Cheetah-Deer*, *Girl*, and *Monkeydog-Monkey* sequences, the JOTS algorithm achieves relative better results only in the region between two red dashed lines. Obviously, the geometric structure of the rigid targets are relative stable. Thus, when the number of superpixels is larger than a certain value, the JOTS algorithm can obtain enough local information of the target to produce desirable results. Meanwhile, for the non-rigid target objects, their geometric structure changes dramatically, which weakens the discriminative power of the location aspect of part model. Thus, if the number of superpixels is too small, the part models are less discriminative to output satisfactory results. On the other hand, if it is too large, the part models will be small and will be removed as the noise during the target motion, which also results in bad performance.

Effectiveness of the regularization term. To demonstrate the effectiveness of the regularization term in the proposed JOTS algorithm, we compare it with two baselines methods, i.e., "JOTS-com" and "w/o regularization" on the SegTrack database using the intersection-over-union overlap metric. The "JOTS-com" method uses only the *Complexity* regularization term in the regularization term (5), i.e., $h_f(\boldsymbol{\mathcal{M}}) = \sum_{i=1}^{k} \mathbb{I}(\exists p : f_p = l_i) \cdot \left(\alpha_7 \cdot h_f^3(\mathcal{M}_i)\right)$, and the

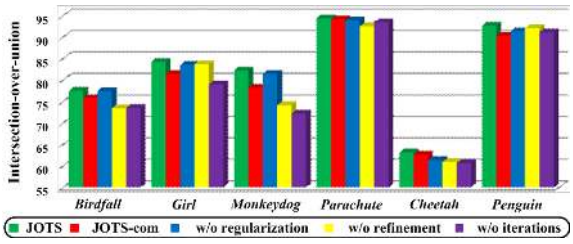

Figure 5. Comparisons of the JOTS algorithm and its four baseline methods of the sequences in the SegTrack database based on the intersection-over-union overlap metric.

"w/o regularization" approach does not use the regularization term in the energy objective (2), i.e., $h_f(\boldsymbol{\mathcal{M}}) = 0$. As shown in Figure 5, the JOTS algorithm performs better than both baseline methods in all sequences, which demonstrates the effectiveness of the well designing regularization term. The regularization term in (6) not only considers the constraints of the number of models (i.e., the *Complexity* regularization term) to remove some useless models, but also the constraints of the size and regularity of the models (i.e., the *Area* and *Profile* regularization terms) to improve the performance (as discussed in Section 4.3).

Integration of multi-part tracking and segmentation. To demonstrate the effectiveness of the integration of multi-part tracking and segmentation in a unified objective function, we construct two baseline methods, i.e., "w/o iterations" and "w/o refinement". The "w/o iterations" method indicates that multi-part tracking and segmentation are not optimized iteratively, and the "w/o refinement" approach indicates that the model refinement step (See Section 5.3) is not included in the iterative process. Figure 5 shows that the JOTS algorithm outperforms both baseline methods in all sequences. Without the iterative process, the multi-part tracking and segmentation modules fail to help each other, and thus the results are less accurate. Without refining hypothesized models, multiple good models can not be added, which reduces the accuracy of the results.

7. Conclusion

In this paper, a joint online tracking and segmentation algorithm based on multi-part models is proposed for online video segmentation. Both multi-part segmentation as pixel labeling and tracking as part models estimation process are integrated in a unified energy minimization formulation, which is effectively solved by a RANSAC-style approach with the α-expansion algorithm. Furthermore, multiple constraints are integrated to regularize the pixel labeling and part models estimation. Extensive experimental results on two benchmark databases demonstrate the effectiveness of the proposed algorithm against the state-of-the-art methods for video segmentation.

Acknowledgment

Longyin Wen, Zhen Lei and Stan Z. Li are supported by the National Natural Science Foundation of China Projects (No.61203267, No.61375037, No.61473291), National Science and Technology Support Program Project (No.2013BAK02B01), Chinese Academy of Sciences Project (No. KGZD-EW-102-2), and AuthenMetric R&D Funds. Dawei Du is supported by the National Natural Science Foundation of China Projects (No.61472388). Ming-Hsuan Yang is supported in part by NSF CAREER Grant (No.1149783) and NSF IIS Grant (No.1152576).

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Ssstrunk. SLIC Superpixels. Technical report, 2010. 2, 3, 5, 6

[2] C. Aeschliman, J. Park, and A. Kak. A probabilistic framework for joint segmentation and tracking. In *CVPR*, pages 1371–1378, 2010. 2

[3] A. Andriyenko, K. Schindler, and S. Roth. Discrete-continuous optimization for multi-target tracking. In *CVPR*, pages 1926–1933, 2012. 3, 4, 5

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *TPAMI*, 23(11):1222–1239, 2001. 3, 5

[5] T. Brox and J. Malik. Large displacement optical flow: Descriptor matching in variational motion estimation. *TPAMI*, 33(3):500–513, 2011. 4

[6] Z. Cai, L. Wen, Z. Lei, N. Vasconcelos, and S. Z. Li. Robust deformable and occluded object tracking with dynamic graph. *TIP*, 23(12):5497–5509, 2014. 2, 3, 6, 7

[7] J. Chang and J. W. Fisher, III. Topology-constrained layered tracking with latent flow. In *ICCV*, 2013. 1, 2, 6, 7

[8] J. Chang, D. Wei, and J. W. F. III. A video representation using temporal superpixels. In *CVPR*, pages 2051–2058, 2013. 2

[9] P. Chockalingam, S. N. Pradeep, and S. Birchfield. Adaptive fragments-based tracking of non-rigid objects using level sets. In *ICCV*, pages 1530–1537, 2009. 2

[10] A. Delong, A. Osokin, H. N. Isack, and Y. Boykov. Fast approximate energy minimization with label costs. *IJCV*, 96(1):1–27, 2012. 3, 4

[11] S. Duffner and C. Garcia. PixelTrack: a fast adaptive algorithm for tracking non-rigid objects . In *ICCV*, pages 2480–2487, Dec. 2013. 2

[12] M. Godec, P. M. Roth, and H. Bischof. Hough-based tracking of non-rigid objects. In *ICCV*, pages 81–88, 2011. 2, 6, 7

[13] M. Grundmann, V. Kwatra, M. Han, and I. A. Essa. Efficient hierarchical graph-based video segmentation. In *CVPR*, pages 2141–2148, 2010. 1, 6, 7

[14] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman. Geodesic star convexity for interactive image segmentation. In *CVPR*, pages 3129–3136, 2010. 2, 6

[15] Z. Hong, C. Wang, X. Mei, D. Prokhorov, and D. Tao. Tracking using multilevel quantizations. In *ECCV*, pages 155–171, 2014. 2

[16] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *ICCV*, pages 1995–2002, 2011. 1, 6, 7

[17] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *ICCV*, pages 1482–1489, 2005. 3

[18] 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, 7

[19] T. Ma and L. J. Latecki. Maximum weight cliques with mutex constraints for video object segmentation. In *CVPR*, pages 670–677, 2012. 1

[20] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstraint video. In *ICCV*, pages 1–6, 2013. 1

[21] A. V. Reina, S. Avidan, H. Pfister, and E. L. Miller. Multiple hypothesis video segmentation from superpixel flows. In *ECCV*, pages 268–281, 2010. 1

[22] X. Ren and J. Malik. Tracking as repeated figure/ground segmentation. In *CVPR*, 2007. 2

[23] C. Rother, V. Kolmogorov, and A. Blake. "GrabCut": Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004. 2

[24] D. Tsai, M. Flagg, and J. M. Rehg. Motion coherent tracking with multi-label mrf optimization. In *BMVC*, pages 1–11, 2010. 1, 5, 6, 7

[25] S. Wang, H. Lu, F. Yang, and M.-H. Yang. Superpixel tracking. In *ICCV*, pages 1323–1330, 2011. 2, 6, 7

[26] T. Wang and J. P. Collomosse. Probabilistic motion diffusion of labeling priors for coherent video segmentation. *TMM*, 14(2):389–400, 2012. 1

[27] D. Zhang, O. Javed, and M. Shah. Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In *CVPR*, pages 628–635, 2013. 1, 6, 7

[28] B. Zhong, X. Yuan, R. Ji, Y. Yan, Z. Cui, X. Hong, Y. Chen, T. Wang, D. Chen, and J. Yu. Structured partial least squares for simultaneous object tracking and segmentation. *Neurocomputing*, 133(10):317–327, 2014. 2