Video Segmentation with Joint Object and Trajectory Labeling

Michael Ying Yang, Bodo Rosenhahn
Institute for Information Processing (TNT), Leibniz University Hannover
Appelstr. 9A, 30167 Hannover, Germany
{yang,rosenhahn}@tnt.uni-hannover.de

Abstract

Unsupervised video object segmentation is a challenging problem because it involves a large amount of data and object appearance may significantly change over time. In this paper, we propose a bottom-up approach for the combination of object segmentation and motion segmentation using a novel graphical model, which is formulated as inference in a conditional random field (CRF) model. This model combines object labeling and trajectory clustering in a unified probabilistic framework. The CRF contains binary variables representing the class labels of image pixels as well as binary variables indicating the correctness of trajectory clustering, which integrates dense local interaction and sparse global constraint. An optimization scheme based on a coordinate ascent style procedure is proposed to solve the inference problem. We evaluate our proposed framework by comparing it to other video and motion segmentation algorithms. Our method achieves improved performance on state-of-the-art benchmark datasets.

1. Introduction

One of the great challenges in computer vision is automatic segmentation of complex dynamic content in videos, so called object segmentation, which is to produce a binary segmentation, separating foreground objects from their background in an unannotated video. This is a challenging task, as local image measurements often provide only a weak cue. Object appearance may significantly change over the frames of the video due to changes in the camera viewpoint, scene illumination or object deformation. In general, segmentation must capture both short range correlations (within a frame and between successive frames) and long range correlations (across many frames) in the video. Object segmentation is the basis for many potential applications including object tracking, object recognition, 3D reconstruction, activity recognition, and video retrieval. Due to its potential applications, there is increasing number of works [17, 20] addressing the problem of video object segmentation in recent years. Many approaches extend single image segmentation techniques to multiple frames, exploiting the fact that there is redundancy along the time axis and that the motion field is smooth. The problems associated with these methods include drift, occlusion, and appearance adaption. Integrating long-term cues in the segmentation process might help solve these problems. In fact, video provides rich additional cues beyond a single image. These cues include object motion, temporal continuity, and long-range temporal object interactions, etc. Motion segmentation exploits these cues, which formulates clustering objectives to group pixels from all frames. However, motion segmentation results are only in discrete and sparse positions available [9].

In this paper, we overcome aforementioned problems by merging image segmentation and motion segmentation. We propose a method to obtain a spatio-temporal foreground segmentation of a video that respects object boundaries, as shown in Fig. 1, and at the same time perform trajectory labeling. Different from previous approaches, we address

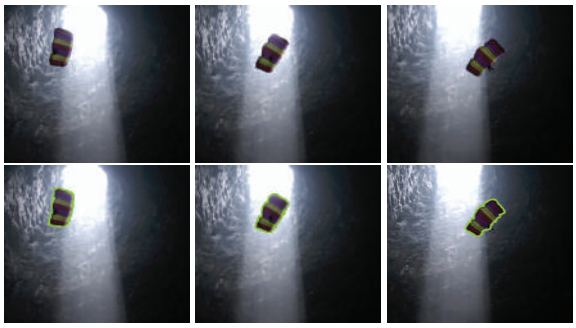

Figure 1. Video object segmentation. Input: unannotated video. Output: Foreground object in each frame.

the foreground segmentation by partitioning frames using a novel graphical model on pixel level, which is dense in spatial domain, yet sparse in temporal domain. We formulate the problem as inference in a conditional random field (CRF). We make use of point trajectories, which have rich grouping information in their motion differences. The CRF

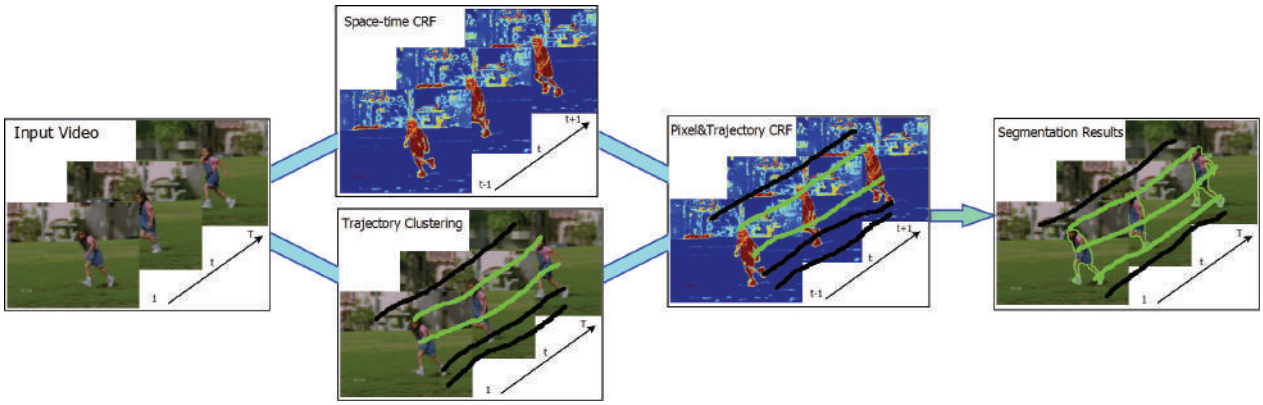

Figure 2. Video segmentation overview. Input: unannotated video. Output: Foreground object segments for all frames (the green boundary overlays with each frame for visualization), and trajectory labeling results. We optimize over pixels and trajectories in the joint space via a space-time CRF: both foreground estimation and trajectory clustering are modeled as energy potentials in the model. Here, the black trajectories are classified as background while the green ones are foreground.

contains binary variables representing the class labels of image pixels as well as binary variables indicating the correctness of trajectory clustering. Joint object and trajectory segmentation is formulated as a pixel and trajectory labeling problem of assigning each pixel and trajectory with either foreground or background. An overview of our proposed method is given in Fig. 2.

Contributions Our main contribution is a fully automatic and unsupervised bottom-up approach for the combination of object segmentation and motion segmentation, which is formulated as inference in a unified CRF model. The CRF contains pixel labeling and trajectory clustering in a single energy function, which integrates dense local interaction and sparse global constraints. We optimize over pixels and trajectories in the joint space via a space-time CRF: both foreground estimation and trajectory clustering are modeled as energy potentials. An optimization scheme based on a coordinate ascent style procedure is proposed to solve the inference problem. To the best of our knowledge, this paper is the first one to combine object labeling and trajectory clustering in a unified probabilistic framework.

The following sections are organized as follows. The related works are discussed in Sec. 2. Section 3 introduces the CRF model for video segmentation and the trajectory clustering. Our proposed approach is described in detail in Sec. 4. In Sec. 5, experimental results are presented. Finally, this work is concluded and future work is discussed in Sec. 6.

2. Related Work

Video object segmentation is often performed in an interactive or supervised manner. Interactive methods require a user to perform object boundary annotation in some key frames, which are then propagated to other frames [24, 32, 31]. Tracking-based methods attempt to reduce the supervision to a manual segmentation on only the first frame [26, 11, 30]. However, all such methods demand user input of drawing regions of interest, therefore not fully automatic, and may suffer from sensitivity to a user's annotation experience.

On the other hand, bottom-up approaches can segment videos in a fully automatic manner, based on cues like motion and appearance. Motion segmentation methods cluster pixels in video using bottom-up motion cues. Recent methods perform pixel-level segmentation in a spatio-temporal video volume from scratch [17], begin with an image segmentation per frame and then match segments across nearby frames [25]. Without any top-down notion of objects, however, such methods tend to over-segment, yielding regions that may lack semantic meaning. [8] attempt to segment objects in video by tracking and splitting/merging image regions. [25] extract multiple segmentation hypotheses in each frame, and then search for a segmentation consistent over multiple frames. Spatio-temporal segmentation of video sequences into segments with coherent local properties has been also addressed by graph-based approaches [17]. However, these methods are limited by the analysis performed at a local level. [20] first discover key-segments and group them to predict the foreground objects in a video. [21] introduce maximum weight cliques with mutex constraints in the region graph to obtain reliable segmentations of foreground object. In this work, we also conduct graph-based segmentation. But additionally, we incorporate long-range motion cues into the segmentation.

Similar to video segmentation, grouping point trajectories in video sequences based on independent motions, so called motion segmentation, has received significant atten-

tion. Recently, impressive results in grouping point trajectories were shown by Brox and Malik [9] who carefully analyze motion differences between pairs of tracks and cluster the resulting affinity matrix using normalized cuts [29]. These sparse trajectory clusters are used in [23] to obtain dense object segmentation. Strong shape priors are derived from a multi-level super-pixel segmentation [2], which preserve the main borders between objects. Super-pixels are labeled and merged using the motion segmentation tracks and a multi-level variational approach. A tracking framework for segmenting objects in crowded scenes is proposed in [15], which mediates grouping cues from two levels of tracking granularities, detection tracklets and point trajectories. [14] propose detecting discontinuities of embedding density between spatially neighboring trajectories. Then Gabriel graph is used for converting trajectory clustering to dense image segmentation. [12] present an approach for motion segmentation using multi-scale clustering of frame-to-frame keypoint correspondences instead of trajectories. Another class of spatio-temporal techniques take advantage of all the frames in a video. They treat the video as a 3D space-time volume [19, 28]. Such large amount of data usually results in expensive computational time. Instead of processing all the frames simultaneously, we make use of point trajectories to segment the successive frames, which all together is dense in space, yet sparse in time. As will be shown in this work, video segmentation benefits from motion segmentation, and vice versa.

3. Preliminaries

We begin by describing the CRF model for video segmentation. We then introduce the clustering technique for point trajectories.

3.1. Video object segmentation

Given a video sequence $\boldsymbol{I} = \{I_t\}$, we formulate video segmentation as a pixel labelling problem of assigning each pixel in frame I_t with either foreground or background. Consider a set of the random variables $\{X_i, i \in \mathcal{V}\}$ defined over an undirected graph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, where X_i is associated with a node $i \in \mathcal{V} = \{1, \ldots, n\}$. The CRF is defined over $\mathcal{H}$, so that each node i corresponds to a pixel p_i and an edge between two nodes corresponds to the cost of a cut between two pixels. Let $\boldsymbol{x} = \{x_i\}$ denote the labeling of the CRF which refers to any possible assignment of labels to the random variables, and takes values from the set $\boldsymbol{L} = \{0, 1\}^n$, where 0 corresponds to background and 1 corresponds to foreground. Its energy function $E(\boldsymbol{x})$ can be written as

$$E(\boldsymbol{x}) = \sum_{i \in \mathcal{V}} \phi_i(x_i) + \alpha \sum_{\{i,j\} \in \mathcal{E}} \phi_{ij}(x_i, x_j) \qquad (1)$$

where ϕ_i and ϕ_{ij} are the unary and pairwise potentials respectively, which both depend on the observed data $\boldsymbol{I}$. α is the weighting coefficient in the model. The edge set $\mathcal{E}$ is commonly chosen to define a 6 neighborhood [25, 20], which consists of 4 spatially neighboring pixels in the same frame, and two temporally neighboring pixels in adjacent frames. We assign a pixel's temporal neighbor in the next frame by its optical flow vector displacement [10]. This energy function, Eq. (1), encourages spatial homogeneity of contrast within each frame and temporal consistency between frames.

The most probable or MAP labelling $\boldsymbol{x}^*$ of the random field can be found by minimizing the energy function $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x} \in \boldsymbol{L}} E(\boldsymbol{x})$. While the exact minimization is generally intractable on general CRF, a good approximation can be found efficiently using graph cut based methods [7] or belief propagation [22].

3.2. Trajectory clustering

Long-term motion can provide strong low-level cues for many vision tasks. For example, two static objects can be separated based on their past or future independent motion if this motion evidence is propagated over time. Video segmentation approaches segment objects following the Gestalt principle of common fate, often enhanced by large temporal context of point trajectories. We define a trajectory tr_r to be a sequence of space-time points: $\text{tr}_r = \{(lx_r^t, ly_r^t), t \in T_r\}$, where T_r is the frame span of tr_r, and (lx, ly) is the pixel location. We obtain point trajectory by tracking pixels across frames using the optical flow [10]. Point trajectories are dense in space and can have various lengths.

Trajectories have rich grouping information in their motion differences. We define pairwise affinities between all trajectories that share at least one frame, yielding the affinity matrix W for the whole sequence. We set affinities $W(\text{tr}_r, \text{tr}_s)$ between trajectories tr_r and tr_s according to the maximum velocity difference v_{rs} computed during their time overlap

$$W(\text{tr}_r, \text{tr}_s) = \exp[-\text{dst}_{rs}(d_{sp}\frac{v_{rs}^2}{\sigma_v^2})] \qquad (2)$$

where dst_{rs} denotes the maximum Euclidean distance between tr_r and tr_s, and σ_v is the normalization factor. Penalizing maximum velocity difference takes advantage of the most informative frames in the time overlap between tr_r and tr_s [9]. d_{sp} denotes the average spatial Euclidean distance of tr_r and tr_s in the common time window. Multiplying with the spatial distance ensures that only proximate points can generate high affinities. We then classify trajectories as foreground or background by performing spectral clustering on the affinity matrix W [9]. An example is shown in Fig. 3.

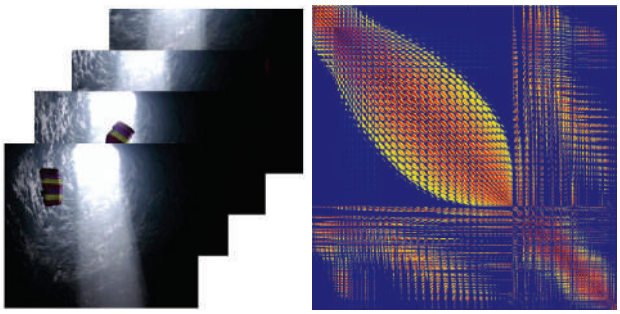

Figure 3. *Left*: an example video sequence. *Right*: corresponding affinity matrix W.

4. Joint object and trajectory segmentation

In this section, we describe our approach to video segmentation. We formulate the problem as inference in a CRF. The random field contains binary variables representing the class labels of image pixels as well as binary variables indicating the correctness of trajectory clustering. The illustration in Fig. 2 gives an overview of our model.

4.1. Formulation

Joint object and trajectory segmentation is formulated as a pixel and trajectory labeling problem of assigning each pixel and trajectory with either foreground or background. Formally, let $x_i \in \{0,1\}$ be a random variable representing the class label of the i-th pixel, while $y_r \in \{0,1\}$ is a random variable associated with the class label of the r-th trajectory. Similar to Eq. (1), the total energy function $E(\boldsymbol{x},\boldsymbol{y})$ for joint segmentation can be written as

$$E(\boldsymbol{x},\boldsymbol{y}) = \sum_{i\in\mathcal{V}} \phi_i(x_i) + \alpha \sum_{\{i,j\}\in\mathcal{E}} \phi_{ij}(x_i,x_j) \qquad (3)$$
$$+ \beta \sum_{\{i,r\}\in\eta} \phi_{ir}(x_i,y_r) + \gamma \sum_{\{r,s\}\in\delta} \phi_{rs}(y_r,y_s)$$

where $\mathcal{V}$ and $\mathcal{E}$ are the sets of nodes and edges in the video frames respectively. η contains all pixel and trajectory pairs that are in correspondence, while the set δ contains all the pairs of trajectories. α,β,γ are the weighting coefficients in the model. ϕ_i is the unary potential encoding the likelihood of pixels belonging to foreground or background. ϕ_{ij} is the pairwise potential, which enforces spatial and temporal consistency between pixels. ϕ_{ir} is the pixel-trajectory compatibility potential, which ensures the corresponding pixel and trajectory take the same label. ϕ_{rs} is the trajectory clustering potential, which encourages foreground and background separation between trajectories. The formulation of these terms will be presented in the remainder of this section.

4.2. Potentials

Unary potentials The unary potential $\phi_i(x_i)$ independently predicts the label x_i based on the frame I_t. The label distribution $\phi_i(x_i)$ is usually calculated by using a classifier. In this paper, we use the Gaussian mixture model (GMM) (i.e. Boykov-Jolly model [5, 27]). GMM is a popular appearance model in object segmentation [3, 16]. The GMM distributions are constructed with a set of simple features, which is a set of pixel colors. Assume a Gaussian mixture with C components, the parameters $\boldsymbol{\theta} = \{\pi_c^f, \mu_c^f, \sigma_c^f, \pi_c^b, \mu_c^b, \sigma_c^b\}_{c=1}^{C}$ are the prior probability, mean, and covariance of the model. Foreground and background trajectories are used for learning these parameters. We set $\phi_i(x_i)$ to be the pixel likelihoods computed from the learned GMM. A pixel that has similar color to the foreground object will have high cost if labeled as background.

Pairwise potentials In segmentation algorithms, spatial and temporal consistencies are usually enforced using pairwise terms based on color difference [27, 20]. ϕ_{ij} is modeled by a standard contrast-dependent function defined in [5, 27], which favors assigning the same label to neighboring pixels with similar color. The edge set $\mathcal{E}$ consists of 4 spatially neighboring pixels in the same frame, and two temporally neighboring pixels in adjacent frames.

Pixel-trajectory compatibility potentials We introduce this pixel-trajectory compatibility term, which imposes a penalty on corresponding pixel and trajectory with different labels. It can be written as

$$\phi_{ir} = 1 - \delta(x_i, y_r) \qquad (4)$$

The corresponding pixel and trajectory pair is determined by whether pixel p_i belongs to tr_r, which defines the set η.

Trajectory clustering potentials We define the trajectory clustering potentials ϕ_{rs} between two trajectories $\mathrm{tr}_r, \mathrm{tr}_s$ as

$$\phi_{rs}(y_r, y_s) = y_r y_s L_{rs} \qquad (5)$$

where L is the Laplacian matrix $L = H^{-1/2} W H^{-1/2}$ [29]. W is the affinity matrix for trajectories defined in Sec. 3.2. H is the diagonal matrix composed of the row sums of W. This term encourages coherent labeling of trajectories. This is equivalent to spectral clustering for all the trajectories in the sequence. Spectral clustering captures essential cluster structure of a graph using the spectrum of the graph Laplacian matrix [29].

4.3. Optimization

The video segmentation problem can be solved by finding the least energy configuration of the CRF defined in

Eq. (3). In general, exact minimization of the energy function E is NP-hard. It is instead solved using approximate algorithms. In our case, minimizing the complex energy function given in Eq. (3), which involves two sets of random variables, is also difficult to approximate. In this paper, we present an optimization scheme based on a coordinate ascent style procedure, alternating between minimizing $E(\boldsymbol{x}, \boldsymbol{y})$ with respect to $\boldsymbol{x}$ for fixed $\boldsymbol{y}$ (1-step) and with respect to $\boldsymbol{y}$ for fixed $\boldsymbol{x}$ (2-step). Convergence to a strong local optimum is usually achieved in 3-4 cycles of iterations. The algorithm is initialized by GMM for pixel labeling and trajectory clustering for trajectory labeling.

1-step For a given binary trajectory labeling $\widehat{\boldsymbol{y}}$, minimizing the total energy function $E(\boldsymbol{x}, \boldsymbol{y})$ in terms of $\boldsymbol{x}$ leads to

$$\min_{\boldsymbol{x}} E(\boldsymbol{x}, \widehat{\boldsymbol{y}}) = \sum_{i \in \mathcal{V}} \phi_i(x_i) + \alpha \sum_{\{i,j\} \in \mathcal{E}} \phi_{ij}(x_i, x_j) + \beta \sum_{\{i,r\} \in \eta} \phi_{ir}(x_i, \widehat{y_r}) \tag{6}$$

When a trajectory labeling is given, the trajectory clustering potentials become constant, and therefore do not affect energy minimization. Furthermore, pixel-trajectory compatibility potentials can effectively be merged to unary potentials. As the pairwise potentials of the energy function in Eq. (6) is a Potts model, it can be minimized using graph cuts [7, 6].

In order to be robust to outliers that may occur due to trajectory clustering errors, we map sparse trajectory points to dense shape-location priors in the pixel-trajectory compatibility potentials. An estimate of the shape, location and scale of the foreground is computed in every frame using a kernel density estimation (KDE) [18] based on the sparse foreground points output by the binary trajectory labeling [13]. The 2D spatial distribution is estimated from the sparse points labeled as foreground (background). The KDE for the object is defined as

$$\hat{f}_h(\boldsymbol{l}) = \frac{1}{\Omega} \sum_{k \in \Omega} K_h(\boldsymbol{l} - \boldsymbol{l}_k) \tag{7}$$

where $\boldsymbol{l}_k$ is the pixel location, Ω is the set of points belonging to the object in that frame, and h is the bandwidth parameter. We use a Gaussian kernel with an automatically adapted bandwidth parameter [4]. This KDE is estimated on sparse points and can be sampled densely to obtain a dense confidence map φ as shown in Fig. 4. This model is highly computationally efficient, similar to the shape priors in [20]. Integrating the confidence map into the energy function in Eq. (6) leads to

$$\min_{\boldsymbol{x}} E(\boldsymbol{x}, \widehat{\boldsymbol{y}}) = \sum_{i \in \mathcal{V}} (\phi_i(x_i) + \beta \varphi_i(x_i)) + \alpha \sum_{\{i,j\} \in \mathcal{E}} \phi_{ij}(x_i, x_j) \tag{8}$$

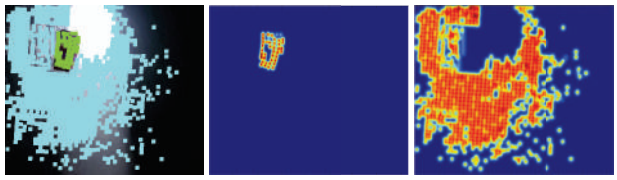

Figure 4. Shape-location prior likelihood. *Left*: sparse label from trajectory clustering [9], *Middle*: foreground confidence map, *Right*: background confidence map.

2-step For a given pixel labeling $\widehat{\boldsymbol{x}}$, minimizing the total energy function $E(\boldsymbol{x}, \boldsymbol{y})$ in terms of $\boldsymbol{y}$ leads to

$$\begin{aligned} \min_{\boldsymbol{y}} E(\widehat{\boldsymbol{x}}, \boldsymbol{y}) &= \beta \sum_{\{i,r\} \in \eta} \phi_{ir}(\widehat{x_i}, y_r) + \gamma \sum_{\{r,s\} \in \delta} \phi_{rs}(y_r, y_s) \\ &= \beta \sum_{r \in R} \phi_r(y_r) + \gamma \sum_{\{r,s\} \in \delta} \phi_{rs}(y_r, y_s) \end{aligned} \tag{9}$$

where R is the set of nodes for the point trajectories. When a pixel labeling is given, the unary and pairwise potentials (first 2 terms in Eq. (3)) become constant. Note that it sometimes happens that the pixel labels $\boldsymbol{x}_k$ along the trajectory tr_r are not consistent. For example, a trajectory consisting of 8 pixel points, which the first 6 are labeled as foreground (1) and the last 2 as background (0). The simple Potts model in Eq. (4) is not a good representative model anymore. We propose the following potentials instead

$$\phi_r(y_r) = \begin{cases} \frac{\mathrm{N}_{\boldsymbol{x}_k=\mathbf{1}}}{|\boldsymbol{x}_k|}, & \text{when } y_r = 1 \\ 1 - \frac{\mathrm{N}_{\boldsymbol{x}_k=\mathbf{1}}}{|\boldsymbol{x}_k|}, & \text{otherwise} \end{cases}$$

where $\mathrm{N}_{\boldsymbol{x}_k=\mathbf{1}}$ is the number of times that the element of $\boldsymbol{x}_k$ is labeled as 1, and $|\boldsymbol{x}_k|$ is the number of elements in $\boldsymbol{x}_k$. As the trajectory clustering potentials ϕ_{rs} are in the forms of Eq. (5), Eq. (9) can also be minimized using graph cuts [7, 6].

5. Experimental Results

5.1. Datasets and implementation details

We present experiments on a number of benchmark sequences, from SegTrack dataset [30] and Berkeley Motion Segmentation Dataset [9], with focus on the *parachute* and *marple3* sequences. The *parachute* sequence from [30] has a spatial resolution of $414\,\mathrm{px} \times 352\,\mathrm{px}$, consists of 51 frames, and per frame pixel-level ground-truth for the primary foreground object. The *marple3* sequence from [9] has a spatial resolution of $350\,\mathrm{px} \times 288\,\mathrm{px}$, consists of 323 frames, and sparse pixel-level ground-truth for the foreground object. The videos span a wide degree of difficulty with challenges such as illumination changes, fg/bg color overlap, large shape deformation, and large camera motion.

Implementation details We use Lab color space histograms with 23 bins per channel, and $C = 5$ component GMMs. To describe motion, we use optical flow histograms with 61 bins per x and y direction, using [10]. For all sequences, point trajectories are obtained by [9], for which there is binary code available. [9] also yields trajectory clusters that look very appealing but are sparse (see Fig. 8 bottom row), for which we use for learning the GMM parameters. For the optimization, we set $\alpha = 5$ for pairwise potentials, $\beta = 0.5$ for pixel-trajectory compatibility potentials, and $\gamma = 5$ for the trajectory clustering potentials. These parameters are fixed for the inference of all sequences. The optimization typically converges in 3 to 4 iterations.

5.2. Results

To quantify segmentation accuracy, we use the average per-frame pixel error rate [30], $\epsilon(S) = \frac{XOR(S,GT)}{F}$, where S is each method's foreground labeling, GT is the ground-truth foreground segmentation, and F is the total number of frames. This score penalizes both over- and under-segmentation. We compare against three state-of-the-art methods: (1) the motion coherence segmentation method [30], (2) the level-set based tracker [11], and (3) the multi-level variational method [23]. First two methods require human labeling of the object boundary in the first frame. Last method requires multi-level superpixel extraction. In contrast, our method requires no hand drawn supervision and no superpixel to guide the segmentation. Table 1 shows the results. Note that segmentation error for the *marple3* sequence is evaluated on the first 50 frames and calculated using the frames where pixel-level ground-truths are available. Our method achieves state-of-the-art results on these sequences. Per-10th-frame pixel label error rate is shown

Table 1. Segmentation error as measured by the average number of incorrect pixels per frame. Lower values are better. We compare our method with three state-of-the-art methods [11, 23, 30].

	Our method	[30]	[11]	[23]
parachute	238	235	502	463
marple3	1610	-	-	2092
Manual seg	no	yes	yes	no

for the *marple3* sequence in Fig. 5. When the parameters (α, β, γ) are set as $(5, 0.7, 5)$, the segmentation error is 1962 for the *marple3* sequence. We also test other parameter combination for the *parachute* sequence, e.g. the segmentation errors are 308 $(1, 0.7, 5)$, 247 $(5, 0.7, 5)$, 270 $(5, 0.6, 5)$, 263 $(5, 0.3, 5)$ respectively, where (α, β, γ) are the different parameter setting. As we use iterative optimization, parameter selection is not critical for the final segmentation results.

Figure 6 and Figure 7 show qualitative segmentation examples [1]. Our method produces high quality segmentations of the foreground object. Fine details and object boundaries are comparable to Ochs and Brox [23]. Furthermore, the stability of the joint object and trajectory segmentation is demonstrated by the improved segmentation over [23]. Ochs and Brox [23] produces only part of the parachute segment from frames 45 to 50 in Fig. 6. While [23] sometimes results in an over-segmentation of an object, our method produces a foreground segmentation at the object-level.

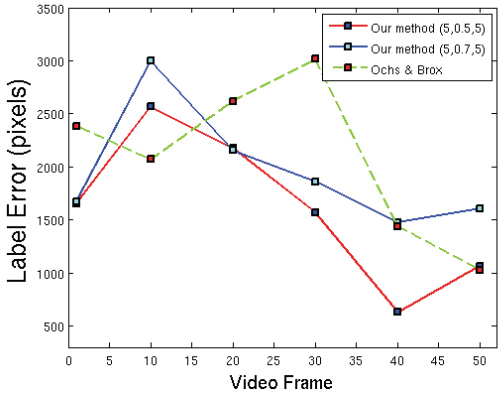

Figure 5. Per-10th-frame pixel label error rate of our approach and [23] for the *marple3* sequence.

As our method jointly optimizes over object pixels and trajectories, we also present the comparison of our trajectory labeling and the trajectory clustering approach [9] in Table 2 in terms of overall clustering error [9]. The overall clustering error is the number of bad labels over the total number of labels on a per-pixel basis. The tool provided by [9] optimally assigns clusters to ground truth regions. The results of our method are consistently better. Motion segmentation on sample frames of the *parachute* sequence is illustrated in Fig. 8. Note that skater was assigned as foreground in trajectory clustering results from [9] (see skater in Fig. 8 2nd and 3rd columns). For our method, during optimization iteration, point trajectories which do not belong to the foreground object has been reassigned as background.

Table 2. Overall clustering error. We compare our method with [9]. Note that we randomly sample ground truth frames of the *parachute* sequence.

	#GT frames	Our method	[9]
marple3	6	1.14	1.18
parachute	6	0.70	0.86
	12	0.70	0.88
	18	0.67	0.85
	24	0.67	0.86

[1] Video sequences are provided at http://www.tnt.uni-hannover.de/project/michael_yang_project.

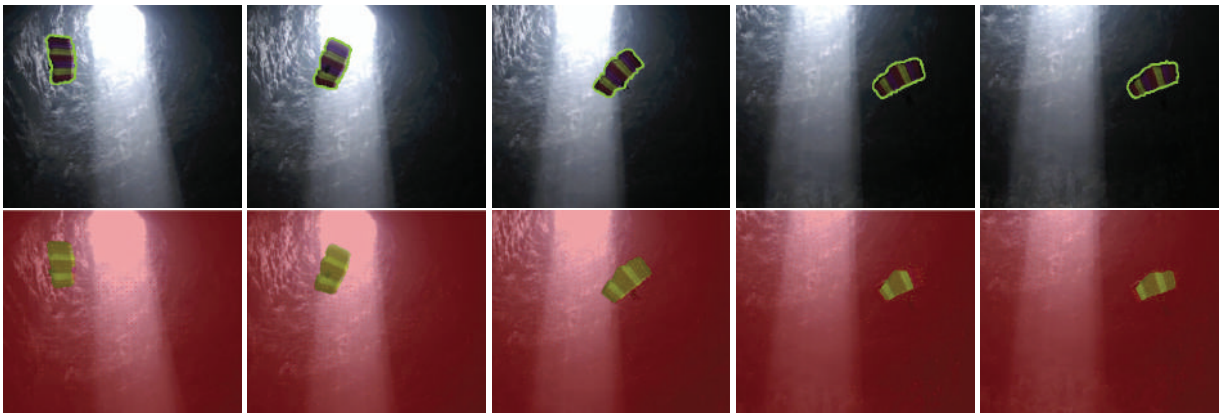

Figure 6. Comparison of our approach and the variational approach [23] on frames 1, 15, 30, 45 and 50 of the *parachute* sequence from the SegTrack dataset [30] (The green boundary overlays with the original image for visualization.). *Top row*: our results, *Bottom row*: Ochs & Brox [23].

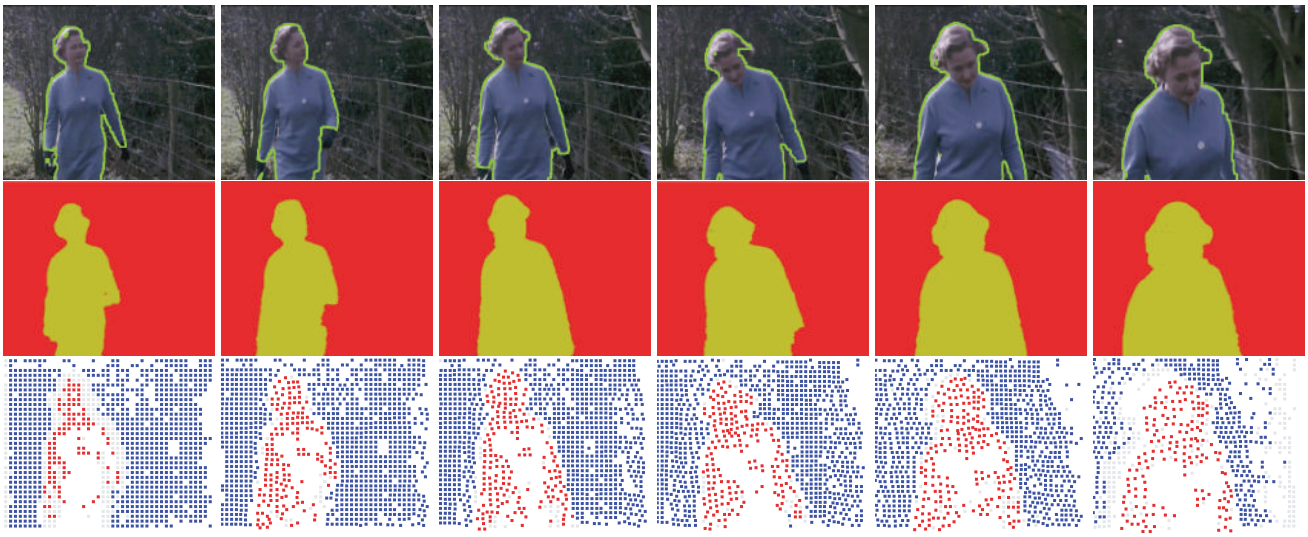

Figure 7. Comparison of our approach and the variational approach [23] on frames 1, 10, 20, 30, 40 and 50 of the *marple3* sequence from the Berkeley Motion Segmentation Dataset [9] (The green boundary overlays with the original image for visualization.). *Top row*: our results, *Middle row*: Ochs & Brox [23], *Bottom row*: motion segmentation results [9].

Figure 9 shows some additional examples that illustrate the final segmentation results of our method on video sequences. The typical failure cases are shown in Fig. 9 bottom row. The failure is usually caused by very bad sparse labeling for GMM initialization. The limitation of our current method is that it relies on good point trajectory clustering results from [9]. This could be alleviated by using *objectness measure* [1] or *key-segments* [20] for GMM initialization.

6. Conclusion

We presented a bottom-up approach for the combination of object segmentation and motion segmentation using a novel CRF model. The CRF contains binary variables representing the class labels of image pixels as well as binary variables indicating the correctness of trajectory clustering, which integrates dense local interaction and sparse global constraints. Hereby, we overcome the limitations of previous bottom-up unsupervised methods that often oversegment an object, and is, to the best of our knowledge, the first approach to combine object labeling and trajectory clustering in a unified probabilistic framework. Our method is fully automatic and unsupervised. The experiments demonstrate the high performance of our approach on benchmark datasets. In our ongoing work, we aim to integrate the proposed model into a system for multi-modal video cosegmentation.

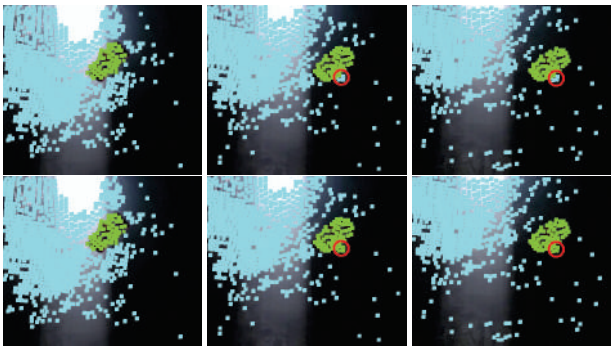

Figure 8. Comparison of our trajectory labeling and the trajectory clustering approach [9] on sample frames of the *parachute* sequence from the SegTrack dataset [30] (see the differences in red circles). *Top row*: our trajectory labeling results, *Bottom row*: trajectory labeling results from [9].

Figure 9. Additional segmentation results.

Acknowledgments

The work is funded by the ERC-Starting Grant (DYNAMIC MINVIP). The authors gratefully acknowledge the support.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *PAMI*, 34(11):2189–2202, 2012. 7

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *PAMI*, 33(5):898–916, 2011. 3

[3] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. Interactively co-segmentating topically related images with intelligent scribble guidance. *IJCV*, 93(3):273–292, 2011. 4

[4] Z. Botev, J. Grotowski, and D. Kroese. Kernel density estimation via diffusion. *Annals of Statistics*, 38(5):2916–2957, 2010. 5

[5] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *ICCV*, pages 105–112, 2001. 4

[6] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, 26:1124–1137, 2004. 5

[7] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23:1222–1239, 2001. 3, 5

[8] W. Brendel and S. Todorovic. Video object segmentation by tracking regions. In *ICCV*, pages 833–840, 2009. 2

[9] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, pages 282–295, 2010. 1, 3, 5, 6, 7, 8

[10] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *PAMI*, 33(3):500–513, 2011. 3, 6

[11] 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, 6

[12] R. Dragon, B. Rosenhahn, and J. Ostermann. Multi-scale clustering of frame-to-frame correspondences for motion segmentation. In *ECCV (2)*, pages 445–458, 2012. 3

[13] L. Ellis and V. Zografos. Online learning for fast segmentation of moving objects. In *ACCV*, 2012. 5

[14] K. Fragkiadaki, G. Zhang, and J. Shi. Video segmentation by tracing discontinuities in a trajectory embedding. In *CVPR*, pages 1846–1853, 2012. 3

[15] K. Fragkiadaki, W. Zhang, G. Zhang, and J. Shi. Two-granularity tracking: Mediating trajectory and detection graphs for tracking under occlusions. In *ECCV (5)*, pages 552–565, 2012. 3

[16] H. Greenspan, J. Goldberger, and A. Mayer. Probabilistic space-time video modeling via piecewise gmm. *PAMI*, 26(3):384–396, 2004. 4

[17] M. Grundmann, V. Kwatra, M. Han, and I. A. Essa. Efficient hierarchical graph-based video segmentation. In *CVPR*, pages 2141–2148, 2010. 1, 2

[18] J.-N. Hwang, S.-R. Lay, and A. Lippman. Nonparametric multivariate density estimation: a comparative study. *IEEE Trans. Sig. Proc.*, 42(10):2795–2810, 1994. 5

[19] A. W. Klein, P.-P. J. Sloan, A. Finkelstein, and M. F. Cohen. Stylized video cubes. In *SIGGRAPH/Eurographics symposium on Computer animation*, pages 15–22, 2002. 3

[20] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *ICCV*, pages 1995–2002, 2011. 1, 2, 3, 4, 5, 7

[21] T. Ma and L. J. Latecki. Maximum weight cliques with mutex constraints for video object segmentation. In *CVPR*, pages 670–677, 2012. 2

[22] K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *UAI*, pages 467–475, 1999. 3

[23] P. Ochs and T. Brox. Object segmentation in video: A hierarchical variational approach for turning point trajectories into dense regions. In *ICCV*, pages 1583–1590, 2011. 3, 6, 7

[24] B. L. Price, B. S. Morse, and S. Cohen. Livecut: Learning-based interactive video segmentation by evaluation of multiple propagated cues. In *ICCV*, pages 779–786, 2009. 2

[25] A. V. Reina, S. Avidan, H. Pfister, and E. L. Miller. Multiple hypothesis video segmentation from superpixel flows. In *ECCV (5)*, pages 268–281, 2010. 2, 3

[26] X. Ren and J. Malik. Tracking as repeated figure/ground segmentation. In *CVPR*, pages 1–8, 2007. 2

[27] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23:309–314, 2004. 4

[28] J. C. Rubio, J. Serrat, and A. M. López. Video co-segmentation. In *ACCV*, pages 1–12, 2012. 3

[29] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 22(8):888–905, 2000. 3, 4

[30] D. Tsai, M. Flagg, A. Nakazawa, and J. M. Rehg. Motion coherent tracking using multi-label mrf optimization. *IJCV*, 100(2):190–202, 2012. 2, 5, 6, 7, 8

[31] S. Vijayanarasimhan and K. Grauman. Active frame selection for label propagation in videos. In *ECCV (5)*, pages 496–509, 2012. 2

[32] J. Yuen, B. C. Russell, C. Liu, and A. Torralba. Labelme video: Building a video database with human annotations. In *ICCV*, pages 1451–1458, 2009. 2