Track to the Future: Spatio-temporal Video Segmentation with Long-range Motion Cues

José Lezama[1] Karteek Alahari[2,3] Josef Sivic[2,3] Ivan Laptev[2,3]

[1]École Normale Supérieure de Cachan [2]INRIA

Abstract

Video provides not only rich visual cues such as motion and appearance, but also much less explored long-range temporal interactions among objects. We aim to capture such interactions and to construct a powerful intermediate-level video representation for subsequent recognition. Motivated by this goal, we seek to obtain spatio-temporal oversegmentation of a video into regions that respect object boundaries and, at the same time, associate object pixels over many video frames. The contributions of this paper are two-fold. First, we develop an efficient spatio-temporal video segmentation algorithm, which naturally incorporates long-range motion cues from the past and future frames in the form of clusters of point tracks with coherent motion. Second, we devise a new track clustering cost function that includes occlusion reasoning, in the form of depth ordering constraints, as well as motion similarity along the tracks. We evaluate the proposed approach on a challenging set of video sequences of office scenes from feature length movies.

1. Introduction

One of the great challenges in computer vision is automatic interpretation of complex dynamic content of videos, including detection, localization, and segmentation of objects and people, as well as understanding their interactions. While this can be attempted by analyzing individual frames independently, video provides rich additional cues not available for a single image. These include motion of objects in the scene, temporal continuity, long-range temporal object interactions, and the causal relations among events. While instantaneous motion cues have been widely addressed in the literature, the long-term interactions and causality remain less explored topics that are usually addressed by high-level object reasoning. In this work, we seek to develop an *intermediate representation*, which exploits long-range temporal cues available in the video, and thus provides a stepping stone towards automatic interpretation of dynamic scenes.

In particular, we aim to obtain a spatio-temporal oversegmentation of video that respects object boundaries, and at the same time temporally associates (subsets of) object pixels whenever they appear in the video. This is a challenging task, as local image measurements often provide only a weak cue for the presence of object boundaries. At the same time, object appearance may significantly change over the frames of the video due to, for example, changes in the camera viewpoint, scene illumination or object orientation. While obtaining a complete segmentation of all objects in the scene may not be possible without additional supervision, we propose to partially address these challenges in this paper.

We combine local image and motion measurements with *long-range motion cues* in the form of carefully grouped point-tracks, which extend over many frames in the video. Incorporating these long point-tracks into spatio-temporal video segmentation brings three principal benefits: (i) pixel regions can be associated by point-tracks over many frames in the video; (ii) locally similar motions can be disambiguated over a larger frame baseline; and (iii) motion and occlusion events can be propagated to frames with no object/camera motion.

The main contributions of this paper are two-fold. First, we develop an efficient spatio-temporal video segmentation algorithm, which naturally incorporates long-range motion cues from past and future frames by exploiting groups of point tracks with coherent motion. Second, we devise a new track grouping cost function that includes occlusion reasoning, in the form of depth ordering constraints, as well as motion similarity along the tracks.

1.1. Related work

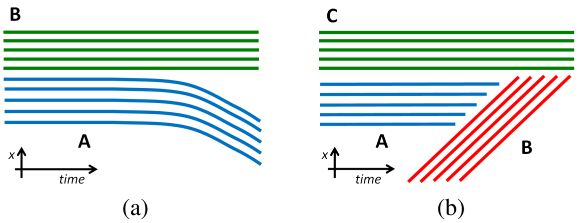
Individual frames in a video can be segmented independently using existing single image segmentation methods [10, 14, 27], but the resulting segmentation is not consistent over consecutive frames. Video sequences can also be segmented into regions of locally coherent motion by analyzing dense motion fields [26, 37] in neighboring frames. Zitnick *et al.* [40] jointly estimate motion and image oversegmentation in a pair of frames. Stein *et al.* [31] analyze

[3]WILLOW project, Laboratoire d'Informatique de l'École Normale Supérieure, ENS/INRIA/CNRS UMR 8548.

local motion and image cues in a small number of neighboring frames to estimate occlusion boundaries [3, 30]. Brendel and Todorovic [8] attempt to segment objects in video by tracking and splitting/merging image regions. Vazquez-Reina *et al.* [34] 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 mean-shift [10] methods [13, 35] or graph-based approaches [16]. However, these methods are limited by the analysis performed at a local level. We build on the hierarchical, graph-based segmentation method of Grundmann *et al.* [16] and extend it by incorporating long-range motion cues into the segmentation.

There has been significant related work on layered representation methods [33, 36, 38], which learn parametric motion and appearance models of video. In this line of research Kumar *et al.* [20] demonstrate detection and tracking of articulated models of walking people and animals, but assume consistent appearance and a locally affine parametric motion model of each object part. In contrast, we are not restricted to such assumptions. Our work is also related to epipolar plane image (EPI) analysis [5], but it focuses on static scenes and constrained camera motions. More recently, Apostoloff and Fitzgibbon [1] assumed only locally linear camera motion, and built an appearance-based detector of spatio-temporal T-junctions with the goal of detecting occluding contours in video. However, they do not address the video segmentation problem. Several high-accuracy interactive video segmentation tools were also developed to target computer graphics applications [2]. It not clear if such methods are easily adaptable to the non-interactive case we consider.

Similar to video segmentation, grouping point trajectories in video sequences based on independent motions has received significant attention. Multi-body factorization approaches [11, 15, 29, 39] have focused on multiple (at least locally) rigidly moving 3D objects under affine camera models. However, complex non-rigid motions, partial and noisy measurements in real-world videos still present a significant challenge. Recently, impressive results in grouping point trajectories were shown by Brox and Malik [9] who carefully analyze motion differences between pairs of tracks and segment the resulting affinity matrix using normalized cuts [27]. We build on this work, but attempt over-segmentation of the video sequence into spatio-temporal regions that respect object boundaries, rather than segment into complete objects, which, we believe, is an under-constrained task, based on motion alone and without additional supervisory signal. In addition, we design a novel track clustering cost function that includes occlusion reasoning and allows recovering partial depth ordering between groups of tracks. Finally, we incorporate the resulting track clusters into an efficient pixel-level video segmenter.



(a) (b)

Figure 1. A toy example illustrating x-t video slices. (a): Objects A and B can be separated early in time if we propagate their motion from a latter point in time. (b): Objects A and C can be separated despite their motion similarity if we take the relative depth ordering, as induced by occlusions, into account. Here, A is occluded by B, and B is occluded by C.

The rest of the paper is organized as follows. Section 2 details the point-track clustering, including the novel occlusion-based cost function. Section 3 builds on the obtained track clusters and describes our extension of the single image segmenter [14] to video sequences using long-range motion cues in the form of clustered point-tracks. Finally, results are presented in Section 4.

2. Track clustering with occlusion reasoning

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 (see Figure 1(a)). Similarly, if an object B occludes A while C occludes B (as shown in Figure 1(b)), we can reason that A and C are in fact two different objects based on the relative depth ordering. Such analyses can also be very useful for higher-level scene understanding in terms of objects and event categories. For example, a point cloud which appears at the door and descends to a chair is very likely to belong to a sitting person who just entered a room.

Our goal in this work is to over-segment the video into groups of pixels belonging to the same object over time and to provide novel building blocks for higher level video interpretation. We start reasoning about the long-term motion by clustering a sparse set of point-tracks extracted from many frames of a video [9]. Our method, however, is not specific to this particular choice and could be used with other point tracking algorithms such as KLT [32] or Particle Video [25].

2.1. Depth order based track clustering

We propose a novel energy-based method for track clustering, which both: (i) groups tracks together based on their similarity, and (ii) establishes a relative depth ordering between track clusters based on the occlusion and disocclusion relations among tracks. As a direct consequence, the method is able to separate tracks, which have similar motion, based on their depth ordering, provided there is sufficient occlusion evidence in the video.

We represent each track with a random variable X_i, which takes a label $x_i \in \mathcal{L}$, representing its cluster assignment. The label set $\mathcal{L} = \{1, 2, \ldots, c\}$, denotes the set of clusters. We select the number of labels manually. However, an automatic selection of number of labels can be addressed with a label cost term, such as in [12]. Let n be the number of tracks in the video sequence, and $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$ be the set of random variables. A labelling $\mathbf{x}$ refers to any possible assignment of labels to the random variables, and takes values from the set $\mathbf{L} = \mathcal{L}^n$. We define the cost of a label assignment $E(\mathbf{x})$, *i.e.* $E(\mathbf{X} = \mathbf{x})$, as follows:

$$E(\mathbf{x}) = \sum_{(i,j)\in\mathcal{E}} \big[\alpha_{ij}\phi_1(x_i, x_j) \quad + \quad (1-\alpha_{ij})\phi_2(x_i, x_j) \\ + \quad \gamma_{ij}\phi_3(x_i, x_j)\big], \quad (1)$$

where $\mathcal{E}$ is the set of pairs of interacting tracks. As detailed later, α_{ij} and γ_{ij} measure, respectively, the motion similarity and occlusion cost for a pair of tracks (i, j). The three potentials, $\phi_1(x_i, x_j)$, $\phi_2(x_i, x_j)$, and $\phi_3(x_i, x_j)$ model the joint cost of labelling tracks i and j. Interacting tracks, $\mathcal{E}$, are only required to have a temporal overlap of at least two frames, *i.e.* any two tracks in the video sequence can potentially interact. This would be equivalent to a generalization of the four or the eight neighborhood used in standard pixel-wise Markov random field models [7]. Note that since our goal is to generate an unsupervised segmentation of a given video sequence, this energy function $E(\mathbf{x})$ does not include any unary terms, which model the cost of assigning labels to each random variable independently. The cost function can be extended to incorporate unary potentials, similar to those in [21, 28].

2.2. Similarity constraints

To assign similar tracks to the same clusters, we use potential $\phi_1(x_i, x_j)$ which takes the form of a standard Potts model [4, 7], *i.e.*,

$$\phi_1(x_i, x_j) = \begin{cases} 0 & \text{if } x_i = x_j, \\ 1 & \text{otherwise.} \end{cases} \quad (2)$$

The intuition for including $\phi_1(\cdot)$ is that two tracks with similar motion and close locations are more likely to belong to the same object, and thus should be assigned the same label. This "attraction" force is modulated by α_{ij} which measures the similarity between two tracks, and ensures that only similar tracks are constrained to take the same label. The values α_{ij} are computed using the distance between the time-corresponding spatial coordinates of track points $\mathbf{a}_i, \mathbf{a}_j$, as well as the distance between time-corresponding velocity values in both tracks $\mathbf{v}_i, \mathbf{v}_j$ as follows:[1]

$$\alpha_{ij} = exp\left(-\frac{(1+||\mathbf{a}_i - \mathbf{a}_j||_2)^2||\mathbf{v}_i - \mathbf{v}_j||_2^2}{2l_{ij}\sigma_s^2}\right), \quad (3)$$

where l_{ij} is the length of the temporal overlap (in frames) between the tracks i and j, and σ_s is a parameter. Note that this track similarity measure is similar to the one used in [9]. Furthermore, the effect of using α_{ij} with $\phi_1(\cdot)$ is equivalent to the contrast-sensitive Potts model [7].

We also use the potential $\phi_2(\cdot)$, which acts as a "repelling" force for tracks that have low similarity values, *i.e.* $(1 - \alpha_{ij})$ is high. This potential is defined as:

$$\phi_2(x_i, x_j) = \begin{cases} 1 & \text{if } x_i = x_j, \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

The potential $\phi_2(\cdot)$ ensures that dis-similar tracks take different labels. The repelling force of $\phi_2(\cdot)$ prevents the trivial solution of all tracks being assigned to the same cluster.

The effect of using $\phi_1(\cdot)$ and $\phi_2(\cdot)$ potentials together with the similarity measure α_{ij} is illustrated in Figure 2(c). As can be seen, minimization of the energy defined in terms of $\phi_1(\cdot)$, $\phi_2(\cdot)$ only separates the tracks of a moving person from the rest of the tracks originating from the static scene.

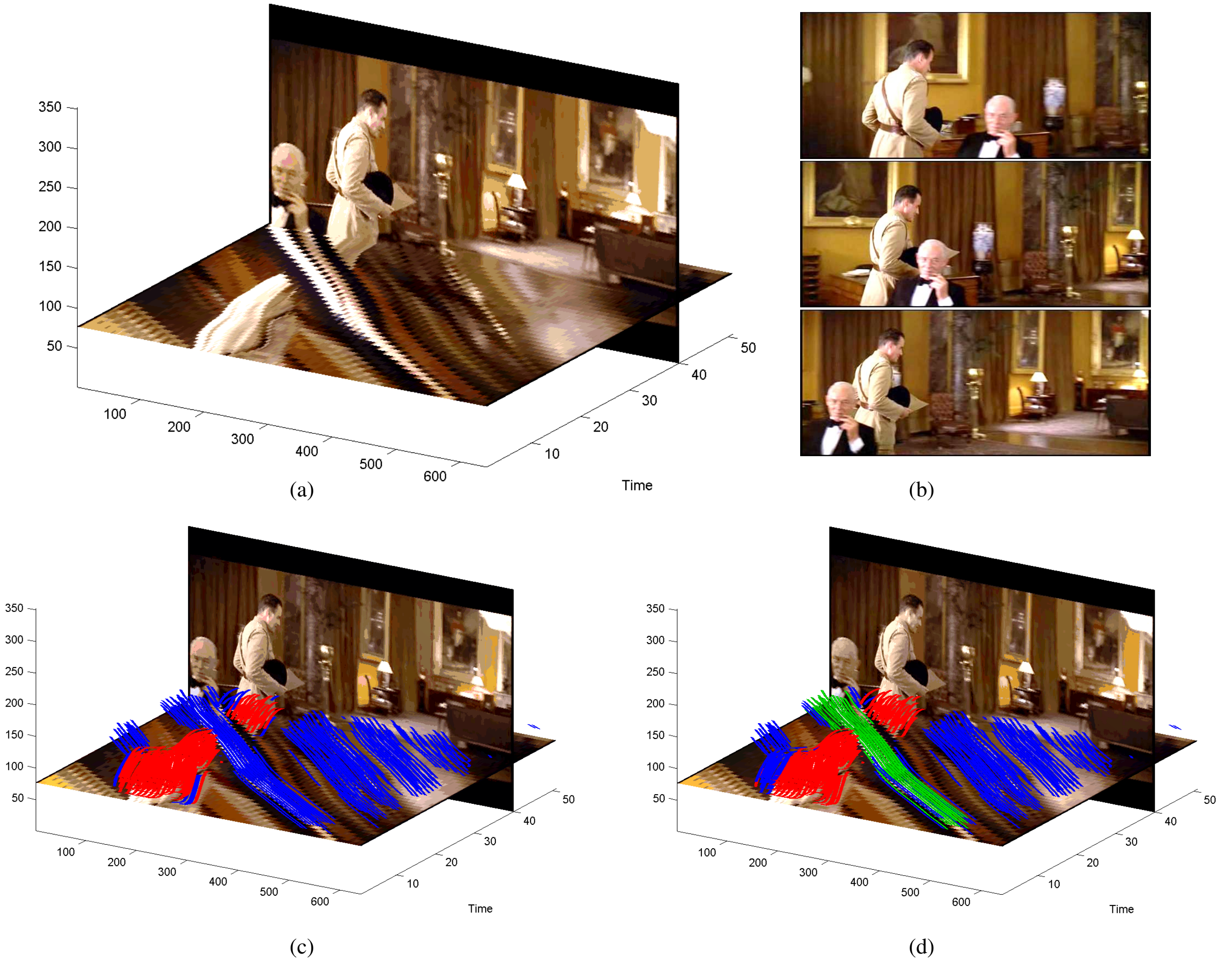
2.3. Depth ordering constraints

Although results in Figure 2(c) are consistent with our expectations, the obtained grouping of tracks is far from being perfect since the method fails to separate different objects sharing similar motion, for example, the static person in the foreground and its background. We address this problem by reasoning about occlusions and disocclusions. We observe that tracks of object A are usually terminated abruptly by the tracks of another object B if B occludes A (we denote this by $B \rightarrow A$). Moreover, $C \rightarrow B \rightarrow A$ provides evidence that C and A most likely belong to different depth layers, *i.e.* $C \rightarrow A$, and therefore C and A should be separated even if they share similar motion (or appearance, or both). Indeed, the case in Figure 1(b) corresponds to the real example in Figure 2, where the background (A) is being occluded by the moving person (B), which in turn is occluded by the sitting person in the front (C). In the following we integrate the notion of occlusion within our clustering framework to enable both improved clustering of tracks and inference of the relative depth ordering between track clusters.

We introduce the potential $\phi_3(x_i, x_j)$ imposing the order on the track labels as follows:

$$\phi_3(x_i, x_j) = \begin{cases} 1 & \text{if } x_i \geq x_j, \\ 0 & \text{if } x_i < x_j. \end{cases} \quad (5)$$

[1] Note that $\mathbf{a}$ is composed of (x, y) coordinates of all points in a track. When comparing two tracks i and j, we use only points from frames where both tracks overlap in time. Similarly, $\mathbf{v}_i, \mathbf{v}_j$ are composed of velocity values $x_t - x_{t-1}$ and $y_t - y_{t-1}$ for frames t and $t-1$ for time-overlapping segments of tracks i and j.



(a)

(b)

(c)

(d)

Figure 2. **Track clustering.** (a)-(b): Illustration of original image sequence with a person passing behind another sitting person. On the space-time slice (a), note the two sets of "T-junctions" generated by the motion boundaries between: (i) the walking person and the background; and (ii) the walking person and the static person in the front. (c) **Results of similarity-based track clustering** using energy function with potentials ϕ_1 and ϕ_2 only (see text). (d) **Results of similarity and occlusion-based track clustering** using the full energy in (1). The introduction of the occlusion potential ϕ_3 to (1) enables separation of the static person (green) from the static background (blue). The correct relative depth ordering of track labels (from back to front: $1 < 2 < 3$) is resolved by occlusion reasoning: 1 – background (blue); 2 – moving person (red); 3 – static person in the front (green). Note that a subset of the original tracks is shown to make the figure readable. A single cluster is obtained for the walking person (red) as there are other tracks which connect the two seemingly different components, *i.e.* a partial occlusion. (Best viewed in color)

This potential is modulated by γ_{ij} in (1) which takes a high value for a pair of tracks (i,j), if j occludes i, *i.e.* $j \rightarrow i$. To measure γ_{ij} from the data, we consider a pairwise asymmetric cost between end-points of track i and all tracks j, close to i. As can be seen from the x-t slice of the video in Figure 2, occlusions often result in "T-junctions" and tracks are terminated by other near-by tracks with different motion. Hence, we define score of γ_{ij} in terms of the difference in velocity $\mathbf{v}_i - \mathbf{v}_j$ and the difference in position D (see Figure 3) between tracks (i,j) at the moment of termination (or start) of a track i as follows:

$$\gamma_{ij} = 1 - exp\left(-\frac{d||\mathbf{v}_i - \mathbf{v}_j||_2^2}{\sigma_o^2}\right), \qquad (6)$$

where σ_o is a parameter and d is a damping coefficient decreasing the occlusion score with the increasing distance D between tracks (i,j): $d = exp(-D^2/\sigma_d^2)$ for some parameter value σ_d. Figure 4 illustrates the computed values of $\gamma_{i\cdot}$ for each end-point of track i maximized over all other tracks j.

We next plug-in the potential $\phi_3(x_i, x_j)$ and the occlusion score γ_{ij} into (1) and optimize the full energy. The resulting clustering of the tracks is illustrated in Figure 2(d). In contrast to result in Figure 2(c), we observe the tracks of

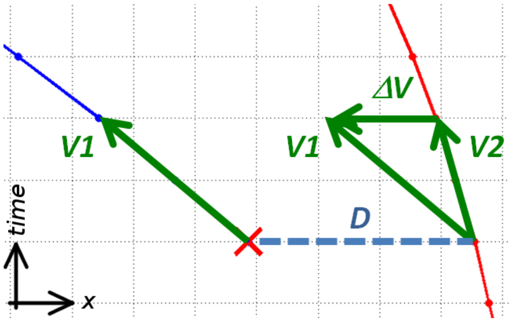


Figure 3. Illustration of measurements involved in the computation of occlusion/disocclusion score γ_{ij} in (6). The velocity vector V_1 at the end-point of the occluded track (blue) is compared to velocity vector V_2 of a near-by track (red). D is the image distance between two tracks.

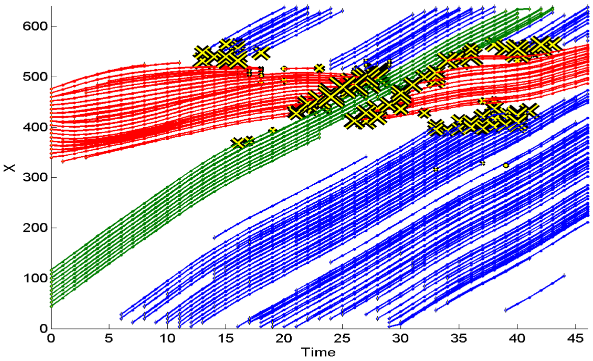


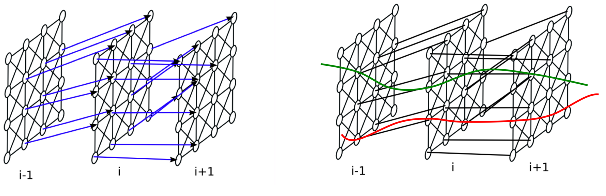
Figure 4. The values of occlusion scores $\gamma_{i\cdot}$ are illustrated for end-points of all tracks extracted on the example video in Figure 2. Large size crosses correspond to large values of $\gamma_{i\cdot}$ and illustrate good prediction of track occlusion in this example.

the sitting person (green) have now been separated from the background (blue). Moreover, the relative depth ordering of the labels has been correctly recovered as green → red → blue. More results for the track clustering as well as for its application to the dense video segmentation are shown in Section 4.

Optimizing the energy function. The most probable or Maximum a Posteriori (MAP) labelling $\mathbf{x}^*$ of the energy function in (1) is defined as: $\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathbf{L}} E(\mathbf{x})$. Note that our energy function does not satisfy the submodularity condition [18] even for the two-label case. Many methods have been proposed to solve such non-submodular energy functions [6, 17, 19]. We use the sequential tree-reweighted message passing (TRW-S) algorithm [17] because of its efficiency and accuracy for our clustering problem. The labels thus obtained for the tracks are then used to define long-range motion cues for video segmentation.

3. Graph-based video segmentation with long-range motion cues

In this section we describe our spatio-temporal video segmentation algorithm, which incorporates long-range motion cues in the form of clusters of point tracks obtained as described in Section 2. Inspired by the graph-based single image segmentation approach of Felzenszwalb and Huttenlocher [14] and its hierarchical extension to video [16], we present an efficient way to incorporate long-range motion cues as additional merging constraints.



(a) (b)

Figure 5. (a) **Extending the graph-based image segmenter of Felzenszwalb and Huttenlocher [14] to video**. Edges following densely estimated motion field connect pixel graphs of neighboring frames, illustrated here for frames $i-1$, i and $i+1$. (b) **Adding long-range motion constraints:** (i) temporal edges connected by point tracks (shown as curves here) receive low cost and hence are merged first; (ii) edges connecting pixels, containing tracks belonging to different groups (shown by two different colors) receive high cost and are not considered for merging.

Review of the graph-based segmentation [14]. An image is represented by a graph $G = (U, E)$ with vertices $u_i \in U$ and edges $(u_i, u_j) \in E$. The elements of U are pixels and the elements of E are the edges linking neighboring pixels. Each edge is assigned a weight, which is a measure of the color dissimilarity between the two pixels linked by that edge. Edges are then ordered by their weight in a non-decreasing order. The ordered list of edges is traversed one by one, deciding if two pixel regions C_1 and C_2 connected by the edge considered are merged according to a score measuring the difference between C_1 and C_2, relative to the internal similarity within C_1 and C_2. Both the region difference and internal similarity are computed from the existing (pre-computed) edges in the graph and no additional measurements in the image are necessary. The algorithm is efficient as its complexity is $O(nlogn)$, where n is the number of edges in the graph.

Extensions to video [16]. The image graph in [14] can be extended to a three-dimensional graph involving all pixels in a video. We follow [16] and add an edge to each pair of neighboring pixels in space and time. For a pixel $(x, y, t)^\top$ we add edges to its 8-connected spatial neighbors as well as to a temporal neighbor $(x + v_x, y + v_y, t + 1)^\top$, where $(v_x, v_y)^\top$ is the velocity vector estimated by optical flow at $(x, y, t)^\top$ (see Figure 5(a)). The weight of an edge between two pixels is set to the Euclidean distance between their color and velocity descriptor vectors $(r, g, b, v_x, v_y)^\top$.

Grundmann *et al.* [16] extend the above pixel graph to a region graph within an iterative hierarchical segmentation framework. At each iteration j the graph is rebuilt by connecting neighboring regions from the previous iteration $j-1$. Edges of the graph have weights corresponding to χ^2-distance between region histogram descriptors of color and motion. We have compared pixel graphs with region graphs and found the latter ones to work better. We therefore build on the the hierarchical segmentation of [16] and extend it as described below.

Incorporating long-range motion cues. Ideally we wish to have a video segmentation where each object is supported by a single spatio-temporal region – an elongated volumetric segment – whose spatial support in each frame, where the object appears, should be the same as the object spatial support. However, segmenting all objects in the video from only low-level cues might be unrealistic without additional top-level supervision [23]. Hence, we aim at producing an over-segmentation, where object's "footprint" in the video is formed from a small set of spatio-temporal regions, which do not cross object boundaries. Preventing the merging of pixels that belong to different objects can be difficult or even impossible to resolve based on only local image and temporal information. In contrast, we introduce long-range information in the form of point-tracks, whose shapes provide, over longer time, additional information for disambiguating pixels with similar local appearance and motion. This is implemented in the following way. First we encourage the created spatio-temporal segments to have a long support in time by building them around point-tracks. Second, we impose constraints on the resulting segmentation based on global track clusters obtained in Section 2.

Encouraging long-range temporal connections. We initialize the hierarchical segmentation by building a three-dimensional pixel graph and add edges that follow point tracks. By giving these edges weight 0, we enforce the merge of all pixels along the track to one region. The tracks then act as "seeds" for the segmentation, which are then "grown" by adding additional pixels based on local similarity in motion and color.

Segmentation with motion constraints. Here we incorporate the track clustering obtained, as described in Section 2, into the segmentation framework. Recall, that tracks are grouped according to their global motion similarity and occlusion constraints into a set of c groups. The situation is illustrated in Figure 5(b). The constraint is incorporated into the segmentation by introducing an additional label for each pixel (node) in the pixel graph. First, pixels along the tracks are labeled according to the cluster to which the track belongs to, a number between 1 and c. The rest of the pixels are labeled -1. Each time the algorithm is going to do a merge, it checks that either the label of the two regions is the same or one of them is -1 and it will only merge if any of these is the case. In case there is a merge, the new region will be labelled with the maximum label of the two regions being merged. This way we make sure that the regions containing tracks from different clusters will never be merged. Note that introducing motion constraints in this form does not affect the complexity of the algorithm.

While the depth ordering constraints are used to produce the track grouping, they affect the segmentation only indirectly in the form of constraints on the track clusters. However, we believe the depth ordering information will be useful for further video understanding tasks, providing useful constraints on object labels, *i.e.* the "wall" should appear behind "a person".

4. Results

Dataset. The dataset used for experiments consists of short video clips taken from the Hollywood 2 dataset [22]. The Hollywood 2 dataset contains scenes from 69 Hollywood movies and provides a challenging set of videos for automatic video interpretation. In order to focus on a manageable but reasonably complex setting, we constrained the video clip selection to scenes taking place in office environments. From this first selection, we chose 10 clips with significant motion as well as occlusions and disocclusions. For each clip, object boundaries were annotated in three selected frames, with a separation of 20 frames between each other. Annotations were made using the LabelMe tool [24] for object classes 'person', 'desk', 'chair', 'lamp', 'cabinet' and 'painting'.

Evaluation method. We aim to have spatio-temporal regions that support objects across all frames in videos. With this intuition, we will measure how well the obtained regions 'follow' the actual object support. For each video, we have annotated three frames (with a gap of 20 frames) with outlines of all considered objects. Then we take one of the three annotated frames, and for each annotated object in this frame, we take the spatio-temporal regions which have a significant spatial overlap (measured by the standard intersection over union score) with the objects in that particular frame. Then we measure how well these selected regions propagate in time, by measuring the overlap of their union with the ground truth object annotations in the two other annotated frames.

Formally, for frame f_i, we compute the following "propagation score":

$$s_i = \frac{1}{|\mathcal{O}|} \sum_{o \in (\mathcal{O})} \left(\frac{M_{GT}(o, f_i) \cap M_S(o, f_i)}{M_{GT}(o, f_i) \cup M_S(o, f_i)} \right), \qquad (7)$$

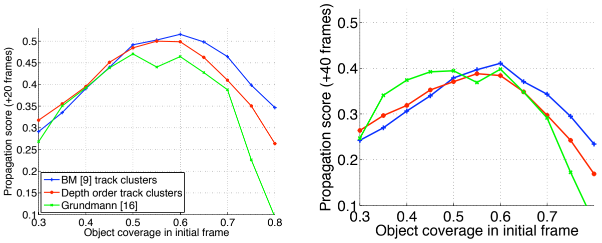
where $\mathcal{O}$ is the set of annotated objects in frame f_i, $M_{GT}(o, f)$ is a binary mask of the annotation of object o

in frame f, $M_S(o, \cdot)$ is the union of volumetric segments that significantly overlap (more than 50%) the annotation of object o in the first frame f_1, and $M_S(o, f)$ is a 2D binary mask resulting from looking at $M_S(o, \cdot)$ in frame f.

Results. Figure 6 shows the average propagation scores (7) obtained when propagating object regions in the 10 video clips over 20 and 40 frames. Each method produces a hierarchy of segmentations with an increasing size of segments. Hence, we plot the propagation score against the object coverage in the first frame measured by the area overlap with the ground truth, given by (7) for $i = 1$. At low hierarchy levels, the (union of) small segments cover the object in the first frame well (object coverage around 0.8), but have low propagation score in time. For medium size segments (medium hierarchy levels) the quality of the segmentation in the first frame decreases (object coverage around 0.6) but the propagation in time is the best. Finally, large segments at high hierarchy levels have very low (< 0.5) object coverage and also their propagation score decreases as they often leak to background and other objects. Results are shown for the proposed video segmenter (Section 3) with tracks clustered either with the method of Section 2 (Depth order track clusters) or the track clustering method of [9] (BM [9] track clusters). Performance is compared with the state-of-the-art video segmentation method of Grundmann *et al.* [16].[2]

The best propagation performance is achieved by our region segmenter in combination with track clustering of [9], outperforming the method of [16] across virtually all reasonable (>0.5) object coverage levels. This demonstrates the advantage of using clustered tracks for spatio-temporal video segmentation. Using the same tracks in combination with our segmentation method in Section 2 gives somewhat worse results, but in contrast to [9], we do not do any post-processing of track clusters, and address a harder clustering task resolving depth ordering in the scene. Qualitative results of video segmentation, object region propagation and track clustering with depth ordering are illustrated in Figures 7 and 8.[3]

Limitations and discussion. Currently the proposed track clustering assumes the relative depth ordering remains constant within the video clip. While, in general, this might not be always true (think of two people circling each other), we observe that this assumption is usually reasonable. Imperfections in tracking can also negatively affect the segmentation results, for example, tracks on the boundary of objects occasionally 'leak' to background and such errors will get propagated to the segmentation. Note also that the relative track ordering is not defined if the tracks do not interact directly or indirectly.



(a) +20 frames

(b) +40 frames

Figure 6. Propagation score vs. the object coverage in the initial frame for the proposed segmentation with long range motion cues (using depth order track clusters or BM [9] track clusters) compared to the segmentation of [16].

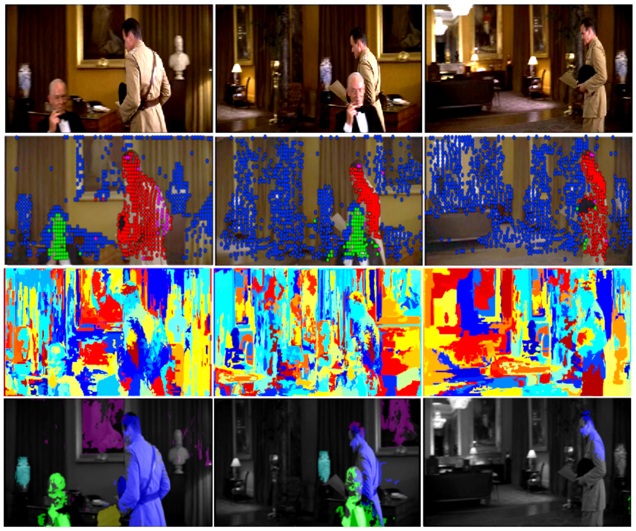


Figure 7. First row: Frames 0, 20 and 40 of the original sequence, respectively. Second row: Tracked points overlaid over the video color-coded according to the obtained clusters. The automatically inferred label ordering (front to back): green, red, magenta, and blue. Note that tracks on the sitting person (green) were correctly separated from background (blue) with the correct depth ordering, despite the fact that the person is not moving. Third row: obtained spatio-temporal over-segmentation of the video frames. Fourth row: the ground truth object regions (left) automatically propagated to the other frames (middle, right).

5. Conclusions

We have developed an efficient spatio-temporal segmentation algorithm incorporating long-range motion cues in the form of groups of point-tracks. Towards this goal, we have devised a new track clustering cost function that includes occlusion reasoning in the form of depth ordering constraints. We are now in a position to build on this representation with the goal of category-level segmentation [21] in video.

[2] We used the authors' implementation available at: `http://www.cc.gatech.edu/cpl/projects/videosegmentation/`

[3] Please see additional results and videos at: `http://www.di.ens.fr/willow/research/videoseg/`

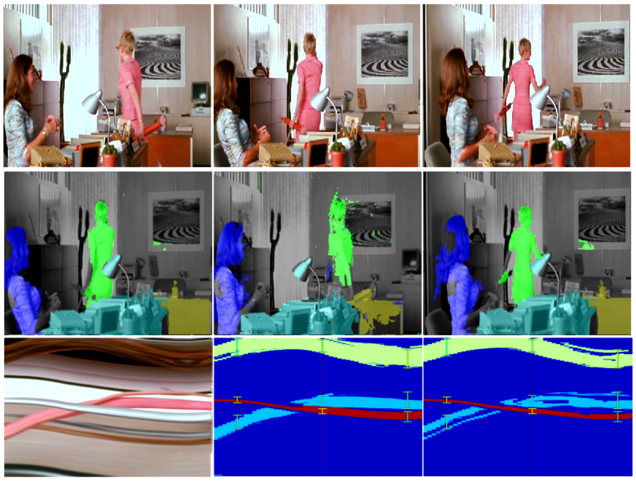


Figure 8. First row: Frames 0, 20 and 40 of the original sequence, respectively. Second row: the ground truth object regions (middle) automatically propagated to the other frames (left, right). Third row: Row 185 (the height of the lamp) of the video shown as a x-t slice (left). The propagated ground truth shown in a x-t slice of the video corresponding to our segmentation with (middle) and without (right) long-range cues. The tracks help to improve the propagation of spatio-temporal regions extended over many frames.

Acknowledgements. This work was partly supported by the Quaero Programme, funded by OSEO, and by the MSR-INRIA laboratory.

References

[1] N. Apostoloff and A. W. Fitzgibbon. Learning spatiotemporal T-junctions for occlusion detection. In *CVPR*, 2005.

[2] X. Bai, J. Wang, D. Simons, and G. Sapiro. Video snapcut: robust video object cutout using localized classifiers. *SIGGRAPH*, 2009.

[3] M. Black and D. Fleet. Probabilistic detection and tracking of motion discontinuities. *IJCV*, 38:231–245, 2000.

[4] A. Blake, C. Rother, M. Brown, P. Perez, and P. H. S. Torr. Interactive image segmentation using an adaptive GMMRF model. In *ECCV*, volume 1, pages 428–441, 2004.

[5] R. C. Bolles, H. H. Baker, and D. H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *IJCV*, 1(1):7–56, 1987.

[6] E. Boros and P. L. Hammer. Pseudo-boolean optimization. *Discrete Appl. Math.*, 123:155–225, 2002.

[7] Y. Y. Boykov and M. P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *ICCV*, 2001.

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

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

[10] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *PAMI*, 2002.

[11] J. P. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *IJCV*, 29(3):159–179, 1998.

[12] A. Delong, A. Osokin, H. N. Isack, and Y. Boykov. Fast approximate energy minimization with label costs. In *CVPR*, 2010.

[13] D. Dementhon. Spatio-temporal segmentation of video by hierarchical mean shift analysis. In *Statistical Methods in Video Processing Workshop (SMVP)*, 2002.

[14] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2):167–181, 2004.

[15] D. Goldman, C. Gonterman, B. Curless, D. Salesin, and S. Seitz. Video annotation, navigation, and composition. In *ACM symposium on User Interface Software and Technology*, 2008.

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

[17] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *PAMI*, 28(10):1568–1583, 2006.

[18] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts. *PAMI*, 26(2):147–159, 2004.

[19] N. Komodakis, Tziritas, and N. Paragios. Fast, approximately optimal solutions for single and dynamic MRFs. In *CVPR*, 2007.

[20] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Learning layered motion segmentations of video. In *ICCV*, 2005.

[21] L. Ladicky, P. Sturgess, K. Alahari, C. Russell, and P. H. S. Torr. What, where & how many? combining object detectors and CRFs. In *ECCV*, 2010.

[22] M. Marszałek, I. Laptev, and C. Schmid. Actions in context. In *CVPR*, 2009.

[23] B. Russell, A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*, 2006.

[24] B. Russell, A. Torralba, K. Murphy, and W. T. Freeman. LabelMe: a database and web-based tool for image annotation. *IJCV*, 2008.

[25] P. Sand and S. Teller. Particle video: Long-range motion estimation using point trajectories. *IJCV*, 80(1), October 2008.

[26] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *ICCV*, 1998.

[27] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 2000.

[28] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *IJCV*, 2009.

[29] J. Sivic, F. Schaffalitzky, and A. Zisserman. Object level grouping for video shots. *IJCV*, 67(2):189–210, 2006.

[30] P. Smith, T. Drummond, and R. Cipolla. Layered motion segmentation and depth ordering by tracking edges. *PAMI*, 2004.

[31] A. Stein, D. Hoiem, and M. Hebert. Learning to extract object boundaries using motion cues. In *ICCV*, 2007.

[32] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical Report CMU-CS-91132, Carnegie Mellon University School of Computer Science, 1991.

[33] P. H. S. Torr, R. Szeliski, and P. Anandan. An integrated bayesian approach to layer extraction from image sequences. *PAMI*, 2001.

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

[35] J. Wang, B. Thiesson, Y. Xu, and M. Cohen. Image and video segmentation by anisotropic kernel mean shift. In *ECCV*, 2004.

[36] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *IEEE Trans. Image Processing Special Issue: Image Sequence Compression*, 3(5):625–638, 1994.

[37] Y. Weiss. Smoothness in layers: motion segmentation using nonparametric mixture estimation. In *CVPR*, 1997.

[38] J. Xiao and M. Shah. Motion layer extraction in the presence of occlusion using graph cuts. *PAMI*, 2005.

[39] J. Yan and M. Pollefeys. A general framework for motion segmentation: independent, articulated, rigid, non-rigid, degenerate and nondegenerate. In *ECCV*, 2006.

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