A Multi-cut Formulation for Joint Segmentation and Tracking of Multiple Objects

Margret Keuper[1], Siyu Tang[2], Yu Zhongjie[2], Bjoern Andres[2]
Thomas Brox[1], Bernt Schiele[2]

[1] Department for Computer Science, University of Freiburg, Germany
[2]Max Planck Institute for Informatics, Saarbruecken, Germany

Abstract. Recently, Minimum Cost Multicut Formulations have been proposed and proven to be successful in both motion trajectory segmentation and multi-target tracking scenarios. Both tasks benefit from decomposing a graphical model into an optimal number of connected components based on attractive and repulsive pairwise terms. The two tasks are formulated on different levels of granularity and, accordingly, leverage mostly local information for motion segmentation and mostly high-level information for multi-target tracking. In this paper we argue that point trajectories and their local relationships can contribute to the high-level task of multi-target tracking and also argue that high-level cues from object detection and tracking are helpful to solve motion segmentation. We propose a joint graphical model for point trajectories and object detections whose Multicuts are solutions to motion segmentation *and* multi-target tracking problems at once. Results on the FBMS59 motion segmentation benchmark as well as on pedestrian tracking sequences from the 2D MOT 2015 benchmark demonstrate the promise of this joint approach.

1 Introduction

Several problems in computer vision, such as image segmentation or motion segmentation in video, are traditionally approached in a low-level, bottom-up way while other tasks like object detection, multi-target tracking, and action recognition often require previously learned model information and are therefore traditionally approached from a high-level perspective.

In this paper, we propose a joint formulation for one such classical high-level problem (multi-target tracking) and a low-level problem (moving object segmentation).

Multi-target tracking and motion segmentation are both active fields in computer vision [1,2,3,4,5,6,7,8,9,10,11,12,13,14]. These two problems are clearly related in the sense that their goal is to determine those regions that belong to the same moving object in an image sequence.

We argue that these interrelated problems can and should be addressed jointly so as to leverage the advantages of both. In particular, the low-level information contained in point trajectories and in their relation to one another

form important cues for the high-level task of multi-target tracking. They carry the information where single, well localized points are moving and can thus help to disambiguate partial occlusions and motion speed changes, both of which are key challenges for multi-target tracking. For motion segmentation, challenges are presented by (1) articulated motion, where purely local cues lead to over-segmentation and (2) coherently moving objects where motion cues cannot tell the objects apart. High level information from an object detector or even an object tracking system is beneficial as it provides information about the rough object location, extent, and possibly re-identification after occlusion.

Ideally, employing such pairwise information between detections may replace higher-order terms on trajectories as proposed in [15]. While it is impossible to tell two rotational or scaling motions apart from only pairs of trajectories, pairs of detection bounding boxes contain enough points to distinguish their motion. With sufficiently complex detection models, even articulated motion can be disambiguated.

To leverage high-level spatial information as well as low-level motion cues in both scenarios, we propose a unified graphical model in which multi-target tracking and motion segmentation are both cast in one graph partitioning problem. As a result, the method provides consistent identity labels in conjunction with accurate segmentations of moving objects.

We show that this joint graphical model improves over the individual, task specific models. Our results improve over the state of the art in motion segmentation evaluated on the FBMS59 [11] motion segmentation benchmark and are competitive on standard multi-target pedestrian tracking sequences [16,6] while additionally providing fine-grained motion segmentations.

2 Related Work

Combining high-level cues with low-level cues is an established idea in computer vision and has been used successfully e.g. for image segmentation [17]. Similarly, motion trajectories have been used for tracking [18,5] and object detections for segmenting moving objects [19]. However, our proposed method is substantially different in that we provide a unified graph structure whose partitioning both solves the low level problem, here, the motion segmentation task, and the high-level problem, i.e. the multi target tracking task, at the same time. In that spirit, the most related previous work is [5], where detectlets, small tracks of detections, are classified in a graphical model that, at the same time, performs trajectory clustering. While we draw from the motivation provided in [5], the key difference to our approach is that we cast both, motion segmentation and multi-target tracking, as clustering problems, allowing for the direct optimization of the Minimum Cost Multicuts [20,21]. Thus, we perform bottom-up segmentation and tracking in a single step.

In [22], tracking and video segmentation are also approched as one problem. However, their appoach employs CRFs instead of Minimum Cost Multicuts, is

built upon temporal superpixels [23] instead of point trajectories and strongly relies on unary terms on these superpixels learned using support vector machines.

In computer vision, Minimum Cost Multicut Formulations have been mainly applied to image segmentation [24,25,26,27]. Exceptiontions are [14] applying this model to motion segmentation and [28] applying it to pedestrian tracking. In [28], Minimum Cost Multicuts have shown to provide a suitable alternative to network flow approaches [29,30]. The clustering nature of minimum cost multicuts avoids the explicit non-maximum suppression step which is a crucial ingredient in disjoint path formulations such as [29,30].

Different approaches towards solving this combinatorial problem of linking the right detection proposal over time use integer linear programming [31,32], MAP estimation [33], or continuous optimization [34]. In these approaches, the complexity usually needs to be reduced by either applying non-maximum suppression or pre-grouping detections into tracklets [2,3,4,5,6,7,8,9].

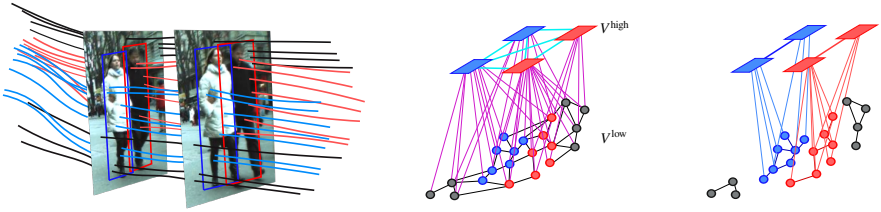
As [35,36,15,37,10,11,12,13,14], we cast motion segmentation as a problem of grouping dense point trajectories. Most of these related approaches employ the spectral clustering paradigm to generate segmentations, while recently [14] have shown the advantages of casting the motion trajectory segmentation as a minimum cost multicut problem.
The approaches of [38,39,35,11,37] base their segmentations on pairwise affinities while [15,40,41] model higher order motions by different means. In our approach, we do not make use of any higher order motion models. In fact, much of the information these terms carry is already contained in the detections we are using, such that we can leverage this information with pairwise terms.

3 Joint Multicut Problem Formulation

Here, we describe the proposed joint high-level - low-level Minimum Cost Multicut Problem formulation which we want to jointly apply to multi-target tracking and moving object segmentation. Our aim is to build a graphical model representing detection and point trajectory nodes and their relationships between one another in a simple, unified way such that the Multicut Problem on this graph directly yields a joint clustering of these high-level and low-level nodes into an optimal number of motion segments and according object tracks.

We define an undirected graph $G = (V, E)$, where $V = \{V^{\text{high}}, V^{\text{low}}\}$ is composed of nodes $v^{\text{high}} \in V^{\text{high}}$ representing high-level entities (detections), and nodes $v^{\text{low}} \in V^{\text{low}}$ representing fine-grained, low-level entities (point trajectories) as depicted in Fig. 1 (b).

To represent the three different types of pairwise relations between these nodes, we define three different kinds of edges. The edge set $E = \{E^{\text{high}}, E^{\text{low}}, E^{\text{hl}}\}$ consists of edges $e^{\text{high}} \in E^{\text{high}}$ defining the pairwise relations between detections (depicted in cyan in Fig. 1 (b)). These can provide pairwise information computed from strong, very specific object features, reflected in the real-valued edge costs $c_{e^{\text{high}}}$. The edges $e^{\text{low}} \in E^{\text{low}}$ represent pairwise relations between point trajectories (depicted in black in Fig. 1 (b)). The according costs $c_{e^{\text{low}}}$ are mostly



(a) high-level and low-level entities (b) proposed graph (c) feasible Multicut

Fig. 1. (a) While pedestrian detections, here drawn as bounding boxes, represent frame-wise high-level information, point trajectories computed on the same sequence represent spatio-temporal low-level cues. Both can be represented as vertices in a joint graphical model (b). The optimal decomposition of this graph into connected components yields both a motion trajectory segmentation of the sequence as well as the tracking of moving objects represented by the detections (c).

based on local information. The edges $e^{\text{hl}} \in E^{\text{hl}}$ represent the pairwise relations between these two levels of granularity (depicted in magenta in Fig. 1 (b)). The Minimum Cost Multicut Problem on this graph defines a binary *edge* labeling problem:

$$\min_{y \in \{0,1\}^E} \sum_{e^{\text{high}} \in E^{\text{high}}} c_{e^{\text{high}}} y_{e^{\text{high}}} + \sum_{e^{\text{low}} \in E^{\text{low}}} c_{e^{\text{low}}} y_{e^{\text{low}}} + \sum_{e^{\text{hl}} \in E^{\text{hl}}} c_{e^{\text{hl}}} y_{e^{\text{hl}}} \qquad (1)$$
$$\text{subject to} \quad y \in \text{MC},$$

where MC is the set of exactly all edge labelings $y \in \{0,1\}^E$ that decompose the graph into connected components. Thus, the feasible solutions to the optimization problem from Eq. 1 are exactly all *partitionings* of the graph G. In the optimal case, each partition describes either the entire backgroud or exactly one object throughout the whole video at two levels of granularity: the tracked bounding boxes of this object and the point trajectories of all points on the object. In Fig. 1 (c), the proposed solution to the Multicut problem on the graph in Fig. 1 (b) contains four clusters: one for each pedestrian tracked over time, and two background clusters in which no detections are contained.

Formally, the feasible set of all multicuts of G can be defined by the cycle inequalities [20] $\forall C \in \text{cycles}(G), \forall e \in C: \quad y_e \leq \sum_{e' \in C \setminus \{e\}} y_{e'}$, making the optimization problem APX-hard [42]. Yet, the benefit of this formulation is that (1) it contains exactly the right set of feasible solutions, and (2) if p_e denotes the probability of an edge $e \in E$ to be cut, then an optimal solution of the Minimum Cost Multicut Problem with the edge weights computed as $c_e = \text{logit}(p_e) = \log \frac{p_e}{1-p_e}$ is a maximally likely decomposition of G. Note that the *logit* function generates real valued costs c_e such that trivial solutions are avoided.

3.1 Pairwise Potentials

In this section, we describe the computation of the pairwise potentials c_e we use in our model. Ideally, one would like to learn terms from training data. However, since the available training datasets for motion segmentation as well as for multi-target tracking are quite small , we choose to rather define intuitive pairwise terms whose parameters have been validated on training data.

Low-level Nodes and Edges In our problem setup, low-level information for motion segmentation and multi-target tracking is built upon point trajectory nodes v^{low} over time and their respective pairwise relations are represented by edge costs $c_{e^{\text{low}}}$.

Low-level Nodes v^{low}: Motion Trajectory Computation A motion trajectory is a spatio-temporal curve that describes the long-term motion of a single tracked point. We compute the motion trajectories according to the method proposed in [11]. For a given point sampling rate, all points in the first video frame having some underlying image structure are tracked based on large displacement optical flow [43] until they are occluded or lost.

The decision about ending a trajectory is made by considering the consistency between forward and backward optical flow. In case of large inconsistencies, a point is assumed to be occluded in one of the two frames. Whenever trajectories end, new trajectories are inserted to maintain the desired sampling rate.

Trajectory Edge Potentials c_e^{low} The edge potentials $c_{e^{\text{low}}}$ between point trajectories v_i^{low} and v_j^{low} are all computed from low-level image and motion information. Motion distances $d^{\text{m}}(v_i^{\text{low}}, v_j^{\text{low}})$ are computed from the maximum motion difference between two trajectories during their common life-time as in [11]. Additionally, we compute color and spatial distances $d^{\text{c}}(v_i^{\text{low}}, v_j^{\text{low}})$ and $d^{\text{sp}}(v_i^{\text{low}}, v_j^{\text{low}})$ between each pair of trajectories with a common life-time and spatial distances for trajectories without temporal overlap as in [14] and combine them non-linearly to $z := c_e^{\text{low}} = \max(\bar{\theta}_0 + \theta_1 d^{\text{m}} + \theta_2 d^{\text{c}} + \theta_3 d^{\text{sp}}, \theta_0 + \theta_1 d^{\text{m}})$. The model parameters θ are set as in [14]. These costs can be mapped to cut probabilities p_e by the logistic function

$$p_e = \frac{1}{1 + \exp(-z)}. \tag{2}$$

High-level Nodes and Edges The high-level nodes v^{high} we consider represent object detections. Since these build upon strong underlying object models, the choice of the object detector is task dependent. While our experiments on the pedestrian tracking sequences make use the Deformable Part Model (DPM) person detector [44], our experiments on the FBMS59 dataset [11] employ a generic object detector (LSDA) [45] which is trained for a wide range of object classes as well as the more specific faster R-CNN [46]. Details on the specific detectors and resulting vertex sets V^{high} are given in the experimental section (Sec. 4).

Detection Edge Potentials c_e^{high} Depending on the employed object detector and the specific task, a variety of different object features could potentially be used to compute high-level pairwise potentials. In our setup, we compute the high-level pairwise terms on simple features based on the intersection over union (IoU) of bounding boxes. On the pedestrian tracking sequences, the high-level part of our graph is built as in [28]. More details for edges in E^{high} will be specified in the experimental section (Sec. 4).

Pairwise Potentials c_e^{hl} between High-level and Low-level Nodes We assume, the safest information we can draw from any kind of object detection represented by a node v_i^{high} is its spatio-temporal center position $\text{pos}_{v_i^{\text{high}}} = (x_{v_i^{\text{high}}}, y_{v_i^{\text{high}}}, t_{v_i^{\text{high}}})^\top$ and size $(w_{v_i^{\text{high}}}, h_{v_i^{\text{high}}})^\top$. Ideally, the underlying object model allows to produce a tentative frame-wise object segmentation or template $T_{v_i^{\text{high}}}$ of the detected object. Such a segmentation template can provide far more information than the bounding box alone. Potentially, a template indicates uncertainties and enables to find regions within each bounding box, where points most likely belong to the detected object. For point trajectory nodes v_j^{low}, the spatio-temporal location $(x_{v_j^{\text{low}}}(t), y_{v_j^{\text{low}}}(t))^\top$ is the most reliable property.

Thus, it makes sense to compute pairwise relations between detections and trajectories according to their spatio-temporal relationship, computed from the normalized spatial distance

$$d^{\text{sp}}(v_i^{\text{high}}, v_j^{\text{low}}) = 2 \left\| \begin{pmatrix} \frac{x_{v_i^{\text{high}}} - x_{v_j^{\text{low}}}(t)}{w_{v_i^{\text{high}}}} \\ \frac{y_{v_i^{\text{high}}} - y_{v_j^{\text{low}}}(t)}{h_{v_i^{\text{high}}}} \end{pmatrix} \right\| \quad \text{for} \quad t = t_{v_i^{\text{high}}} \tag{3}$$

and the template value at the trajectory position $T_{v_i^{\text{high}}}(x_{v_j^{\text{low}}}(t), y_{v_j^{\text{low}}}(t))$. If a trajectory passes through a detected object in a frame t, it probably belongs to that object. If it passes far outside the objects bounding box in a certain frame, it is probably not part of this object.

Thus, we compute edge cut probabilities $p_{e^{\text{hl}}}$ from the above described measures as

$$p_{e_{ij}^{\text{hl}}} = \begin{cases} 1 - T_{v_i^{\text{high}}}(x_{v_j^{\text{low}}}(t), y_{v_j^{\text{low}}}(t)), & \text{if } T_{v_i^{\text{high}}}(x_{v_j^{\text{low}}}(t), y_{v_j^{\text{low}}}(t)) > 0.5 \\ 1, & \text{if } d^{\text{sp}}(v_i^{\text{high}}, v_j^{\text{low}}) > \sigma \\ 0.5, & \text{otherwise} \end{cases} \tag{4}$$

using an application dependent threshold σ.

3.2 Solving Minimum Cost Multicut Problems

The Minimum Cost Multicut problem defined by the integer linear program in Eq. (1) is APX-hard [42]. Still, instances of sizes relevant for computer vision can potentially be solved to optimallity or within tight bounds using branch and cut

[25]. However, finding the optimal solution is not necessary for many real world applications. Recently, the primal heuristic proposed by Kernighan and Lin [47] has shown to provide very reasonable results on image and motion segmentation tasks [27,14]. Alternative heuristics were in [48,49]. In our experiments, we employ [47] because of its computation speed and robust behavior.

4 Experiments

We evaluate the proposed Joint Multicut Formulation on both motion segmentation and multi-target tracking applications. First, we show our results on the FBMS59[11] motion segmentation dataset containing sequences with various object categories and motion patterns. Then, tracking *and* motion segmentation performance will be evaluated on three standard multi-target pedestrian tracking sequences. Last, we evaluate the tracking performance on the 2D MOT 2015 benchmark [50].

4.1 Motion Segmentation Dataset

The FBMS59[11] motion segmentation dataset consists of 59 sequences split into a training set of 29 and a test set of 30 sequences. The videos are of varying length (19 to about 500 frames) and show diverse types of moving objects such as cars, persons and different types of animals.

To exploit the Joint Multicut model for this data, the very first question is how to obtain reliable detections in a video sequence without knowing the category of the object of interest. Here, we evaluate on detections from two different methods : Large Scale Detection through Adaptation (LSDA) [45] and the Faster R-CNN [46].

Large Scale Detection through Adaptation The LSDA is a general object detector, trained to detect 7602 object categories [45]. In our experiments, we directly use the code and model deployed with their paper. It operates on a set of object proposals, which is produced by selective search [53]. The selective search method operates on hierarchical segmentations, which means that we obtain a segmentation mask for each detection bounding box. This segmentation provides a rough spatial and appearance estimation of the object of interest.

To better capture the moving objects in the video, we additionally generate selective search proposals from optical flow images and pass them to the LSDA framework. Example results for the detections and according frame-wise segmentations are given in Fig. 2 (top).

Faster R-CNN Faster R-CNN is an object detector that integrates a region proposal network with the Fast R-CNN [54] network. It achieves state-of-the-art object detection accuracy on several benchmark datasets including PASCAL VOC 2012 and MS COCO with only 300 proposals per image [46]. In our experiments, we directly used the code and model deployed with their paper.

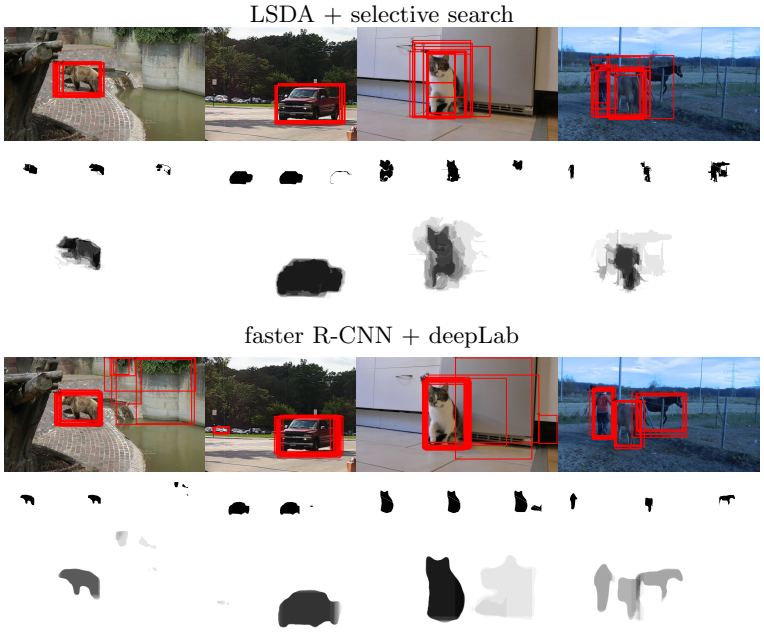
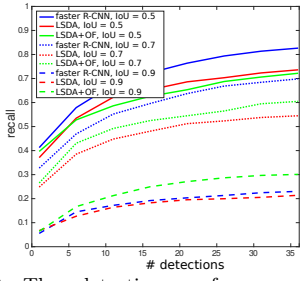


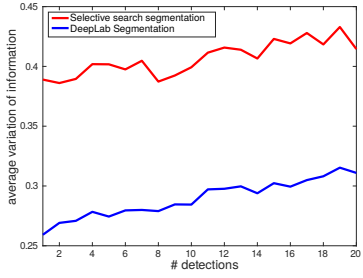
Fig. 2. Examples of the object detections and according segmentations. Top: LSDA detections on images from FBMS59 sequences [11]. The first row shows the best 20 detections. The second row shows three exemplary selective search proposals and third row visualizes the average segmentation of all proposals. Bottom: The corresponding faster R-CNN detections. The first row shows the best 20 detections with a minimum detection score of 0.2. The second row shows three exemplary segmentations from deepLab [51,52] on these detections and third row visualizes the average segmentation.

On the detections, we generate segmentation proposals using DeepLab [51,52], again by directly using their implementation. Example results for the detections and according frame-wise segmentations are given in Fig. 2 (bottom).

Evaluation Fig. 3(a) shows the achieved recall over the number of detections for LSDA [45] and faster R-CNN [46] for different thresholds on the intersection over union (IoU) on the FBMS59 [11] training set. For the higher thresholds, the performance of LSDA is improved when proposals from optical flow images are used (LSDA+OF) and for IoU ≥ 0.9, this setup yields best recall. However, for smaller IoU thresholds, faster R-CNN yields highest recall even without considering optical flow. The comparison of the segment mask proposals from selective search (for LSDA) and deepLab (for faster R-CNN) (Fig. 3 b(b)) shows the potential benefit of DeepLab. The visual comparison on the examples given in Fig. 2 shows that the selective search segmentation proposals selected by LSDA are more diverse than the DeepLab segmentations on the faster R-CNN detection. However, the overall localization quality is worse. We further evaluate detections from both methods in the Joint Multicut model.



(a) The detection performance of LSDA [45] and faster R-CNN [46]. We compare the recall for three different IoU thresholds 0.9, 0.7, and 0.5.



(b) The Variation of Information for the proposed object masks over the number of detections (lower is better).

Fig. 3. Evaluation of the detection and segment proposals on the annotated frames of the FBMS59 [11] training set.

Implementation Details In our graphical model, high-level nodes represent detections from either of the above described methods. For both detectors, we use the same setup. First, we select the most confident detections [1]. From those, we discard some detections according to the statistics of their respective segmentations. Especially masks from the selective search proposals sometimes only cover object outlines or leak to the image boundaries. Thus, if such a mask covers less than 20% of its bounding box or more than 60% of the whole image area, the respective detections are not used as nodes in our graph.

The pairwise terms between detections are computed from the IoU and the normalized distances d^{sp} of their bounding boxes

$$d^{\text{sp}}(v_i^{\text{high}}, v_j^{\text{high}}) = 2 \left\| \begin{pmatrix} \frac{x_{v_i^{\text{high}}} - x_{v_j^{\text{high}}}}{w_{v_i^{\text{high}}} + w_{v_j^{\text{high}}}} \\ \frac{y_{v_i^{\text{high}}} - y_{v_j^{\text{high}}}}{h_{v_i^{\text{high}}} + h_{v_j^{\text{high}}}} \end{pmatrix} \right\|,$$

where $\text{pos}_{v_i^{\text{high}}}$, $w_{v_i^{\text{high}}}$, and $h_{v_i^{\text{high}}}$ are defined as in Eq. (3). For all pairs of detections within one frame and in neighboring frames, the pseudo cut probability is computed as

$$p_{e_{ij}^{\text{high}}} = \begin{cases} 1 - \frac{1}{1+\exp(20*(0.7-\text{IoU}(v_i^{\text{high}}, v_j^{\text{high}})))}, & \text{if IoU}(v_i^{\text{high}}, v_j^{\text{high}}) > 0.7 \\ \frac{1}{1+\exp(5*(1.2-d^{\text{sp}}(v_i^{\text{high}}, v_j^{\text{high}})))}, & \text{if } d^{\text{sp}}(v_i^{\text{high}}, v_j^{\text{high}}) > 1.2 \\ 0.5, & \text{otherwise} \end{cases} \quad (5)$$

The parameters have been set such as to produce reasonable results on the FBMS59 training set. Admittedly, parameter optimization on the training set might further improve our results.

[1] above a threshold of 0.47 for LSDA and 0.97 for faster R-CNN - on a scale between 0 and 1.

	Training set (29 sequences)				**Test set** (30 sequences)			
	P	R	F	O	P	R	F	O
SC [11]	85.10%	62.40%	72.0%	17/65	79.61%	60.91%	69.02%	24/69
SC + Higher Order [15]	81.55%	59.33%	68.68%	16/65	82.11%	64.67%	72.35%	27/69
MCe [14]	**86.73%**	73.08%	79.32%	**31/65**	**87.88%**	67.7%	76.48%	25/69
MCe + det. (LSDA)	86.43%	75.79%	80.7617%	31/65	-	-	-	-
JointMulticut (LSDA)	86.43%	75.79%	80.7634%	31/65	87.46%	70.80%	78.25%	29/69
MCe + det. (f. R-CNN)	83.46%	79.46%	81.41%	35/65	-	-	-	-
JointMulticut (f. R-CNN)	84.85%	**80.17%**	**82.44%**	**35/65**	84.52%	**77.36%**	**80.78%**	**35/69**

Table 1. Results on the FBMS-59 dataset on training (left) and test set (right). We report **P**: average precision, **R**: average recall, **F**: F-measure and **O**: extracted objects with F $\geq$ 75%. All results are computed for sparse trajectory sampling at 8 pixel distance.

The pairwise terms $c_{e^{\text{hl}}}$ are computed from $p_{e^{\text{hl}}}$ as defined in Eq. (5) with $\sigma = 2$. This large threshold accounts for the uncertainty in the bounding box localizations.

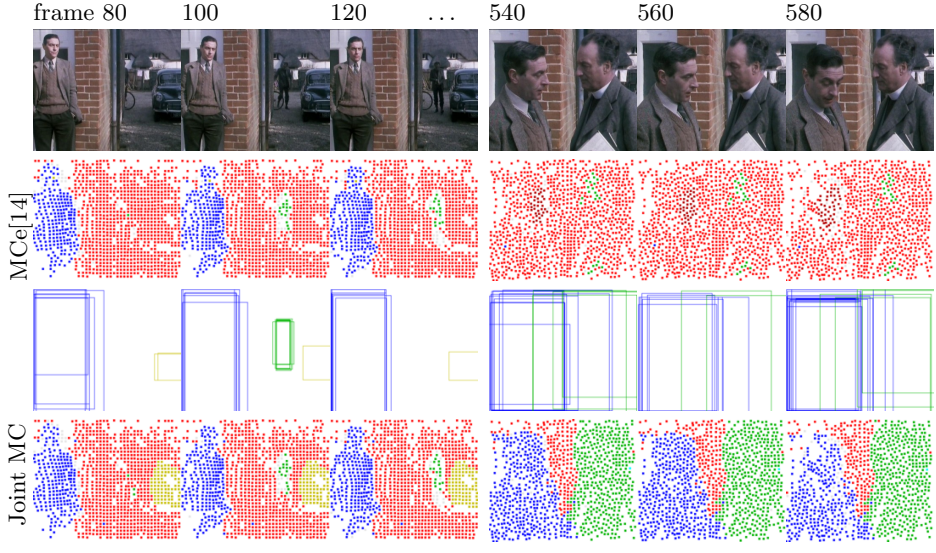


Fig. 4. Comparison of the proposed Joint Multicut model and the multicut on trajectories (MCe) [14] on the *marple6* sequence of FBMS59. While with MCe the segmentation breaks between the shown frames, the tracking information from the bounding box subgraph helps our joint model to segment the two men throughout the sequence. Additionally, static, consistently detected objects like the car in the first part of the sequence are segmented as well. As these are not annotated, this causes over-segmentation on the FBMS59 benchmark evaluation.

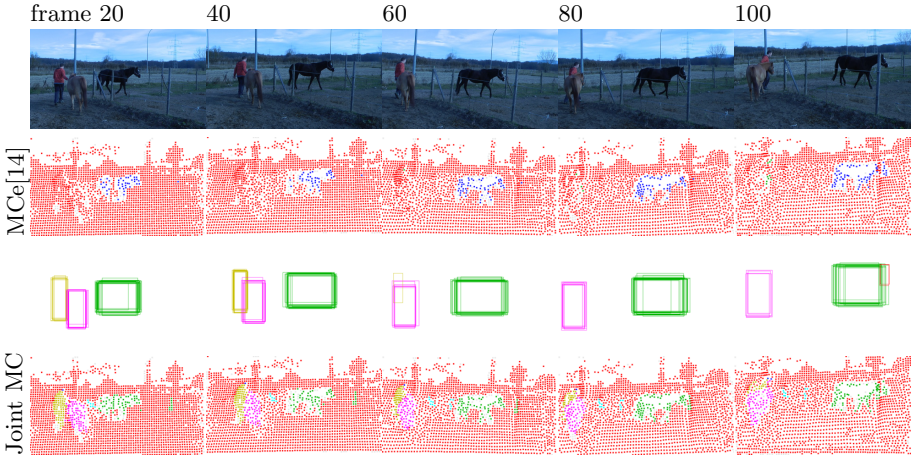


Fig. 5. Comparison of the proposed Joint Multicut model and the multicut on trajectories (MCe) [14] on the *horses06* sequence of FBMS59.

Results Our results are given in Tab. 1. The motion segmentation considering only the trajectory information from [14] performs already well on the FBMS59 benchmark. However, the Joint Multicut model improves over the previous state of the art for both types of object detectors. Note that not only the baseline method of [14] is outperformed with quite a margin on the test set - also the motion segmentation based on higher-order potentials [15] can not compete with the proposed joint model.

To assess the impact of the joint model components, we evaluate not only the full model but also its performance if pairwise terms between detection nodes are omitted (denoted by MCe + detections). For LSDA detections, this result is pretty close to the Joint Multicut model, implying that the pairwise information we currently employ between the bounding boxes is quite weak. However, for the better localized faster R-CNN detections, the high-level pairwise terms contribute significantly to the overall performance of the joint model.

Qualitative examples of the motion segmentation and object tracking results using the faster R-CNN detections are given in Fig. 4 and 5. Due to the detection information and the repulsive terms between those object detections and point trajectories not passing through them, static objects like the car in the *marple6* sequence (yellow cluster) can be segmented. The man approaching the camera in the same sequence can be tracked and segmented (green cluster) throughout the sequence despite the scaling motion. Similarly, in the *horses* sequence, all three moving objects can be tracked and segmented through strong partial occlusions.

Since the ground truth annotations are sparse and only contain moving objects, this dataset was not used to quantitatively evaluate the multi-target tracking performance.

4.2 Multi-Target Tracking Data

First, we evaluate the tracking performance of our Joint Multicut model on the publicly available sequences: TUD-Campus, TUD-Crossing [16] and ParkingLot [6]. These sequences have also been used to evaluate the Subgraph Multicut method [28] and therefore allow for direct comparison to the only previous multicut approach to multi-target tracking.

To assess the quality of the motion segmentation of the joint approach, we annotated the sequences TUD-Campus and ParkingLot with ground truth segmentations of all pedestrians in every 20th frame. For the TUD-Crossing sequence, such annotations have been previously published by [55].

Implementation Details To allow for direct comparison to [28], we compute all high-level information, i.e. the detection nodes $v^{\text{high}} \in V^{\text{high}}$, edges $e^{\text{high}} \in E^{\text{high}}$, and their costs $c_{e^{\text{high}}}$ exactly as reported in [28] with only one difference: the Subgraph Multicut models from [28] employs not only pairwise but also unary terms which our proposed Joint Multicut model does not require. We omit these terms.

In [28], DPM-based person detections [44] are used. To add robustness and enable the computation of more specific pairwise terms, these detections are grouped to small, overlapping tracklets of length 5 as in [16] without applying any Non-Maximum Suppression. Since tracklets are computed in every frame, the same detections can be part of several (at most 5) tracklets.

Pairwise terms between the tracklets are computed from temporal distances, normalized scale differences, speed, spatio-temporal locations and dColorSIFT features [56], combined non-lineraly as in [28].

To compute pairwise terms $c_{e_{ij}^{\text{hl}}}$ between trajectory and tracklet nodes as described in Sec. 3.1, we compute the average pedestrian shape from the shape prior training data provided in [57] (see Fig. 6 (a)). For every detection bbx_k, T_{bbx_k} denotes the pedestrian template shifted and scaled to the kth bounding box position and size. The tracklet information allows to determine the walking direction of the pedestrian, such that the template can be flipped accordingly. For every detection bbx_k with $k = \{1, \ldots, 5\}$ of a tracklet v_i^{high}, the cut propability $p(\text{bbx}_k, v_j^{\text{low}})$ to a trajectory node v_j^{low} is computed according to Eq. (5) with $\sigma = 1.2$.

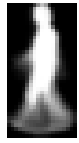
A trajectory node v_j^{low} is linked to a tracklet node v_i^{high} coexisting in a common frame with an edge cost

$$c_{e_{ij}^{hl}} = \sum_{k=1}^{5} \text{logit}(p(\text{bbx}_k, v_j^{\text{low}})). \tag{6}$$

Fig. 6 (b) visualizes the edges between tracklets and point trajectories.

Evaluation Metrics The pedestrian motion segmentation is evaluated with the metrics precision (P), recall (R), f-measure(F) and number of retrieved objects (O) as proposed for the FBMS59 motion segmentation benchmark [11]. To evaluate the tracking performance, we use standard CLEAR MOT as evalua-

(a) Mean pedestrian shape template



(b) Trajectory-Tracklet edges: For every pair of trajectories and tracklets, an edge is inserted if the trajectory either hits a bounding box template or passes sufficiently far outside the bounding box.

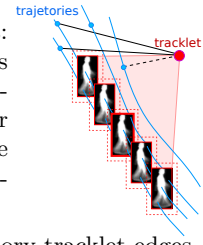


Fig. 6. The average pedestrian shape template and the trajectory-tracklet edges.

	D	P	R	F	O
TUD-Campus					
SC [11]	0.80%	71.67%	47.28%	56.97%	2/8
MCe [14]	0.80%	58.94%	59.68%	59.31%	3/8
MCe + det.	0.80%	73.93%	58.67%	65.43%	3/8
Tracklet MC + traj.	0.80%	**85.15%**	**66.40%**	**74.61%**	**5/8**
JointMulticut	0.80%	75.67%	61.97%	68.13%	3/8
TUD-Crossing					
SC [11]	0.85%	**67.92%**	20.16%	31.09%	0/15
MCe [14]	0.85%	43.78%	38.53%	40.99%	1/15
MCe + det.	0.85%	67.34%	57.33%	61.93%	3/15
Tracklet MC + traj.	0.85%	63.10%	57.45%	60.14%	5/15
JointMulticut	0.85%	62.10%	**64.65%**	**63.35%**	**8/15**
ParkingLot					
SC [11]	1.01%	77.06%	14.79%	24.81%	0/15
MCe [14]	1.01%	58.97%	18.14%	27.75%	0/15
MCe + det.	1.01%	74.51%	63.52%	68.47%	5/15
Tracklet MC + traj.	1.01%	**77.54%**	53.16%	63.01%	4/15
JointMulticut	1.01%	72.62%	**66.93%**	**69.66%**	**7/15**

Table 2. Motion Segmentation on the Multi-Target Tracking sequences. We report **D**: average region density, **P**: average precision, **R**: average recall, **F**: F-measure and **O**: extracted objects with F $\geq$ 75%. All results are computed for sparse trajectory sampling at 8 pixel distance.

tion metrics. Additionally, we report mostly tracked (MT), partly tracked (PT), mostly lost (ML) and fragmentation (FM).

Results The evaluation of the motion segmentations on these three pedestrian tracking sequences produced by the Joint Multicut model is given in Tab. 2. To assess the importance of the model parts, we not only evaluate the full Joint Model but also the performance of the Multicut formulation when not considering pairwise terms between trajectories (Tracklet MC + traj.) as well as the performance when omitting the pairwise terms between tracklet nodes (MCe + det.).On the important f-measure and the number of segmented object, the proposed Joint Multicut model improves over the previous state-of-the-art in motion segmentation on the pedestrian sequences.

Quantitative results on the pedestrian tracking task are given in Tab. 3. Again, we evaluate the importance of the model parts (denoted by MCe + det. and Tracklet MC + traj.). The comparison confirms that the full, joint model performs better than any of its parts.

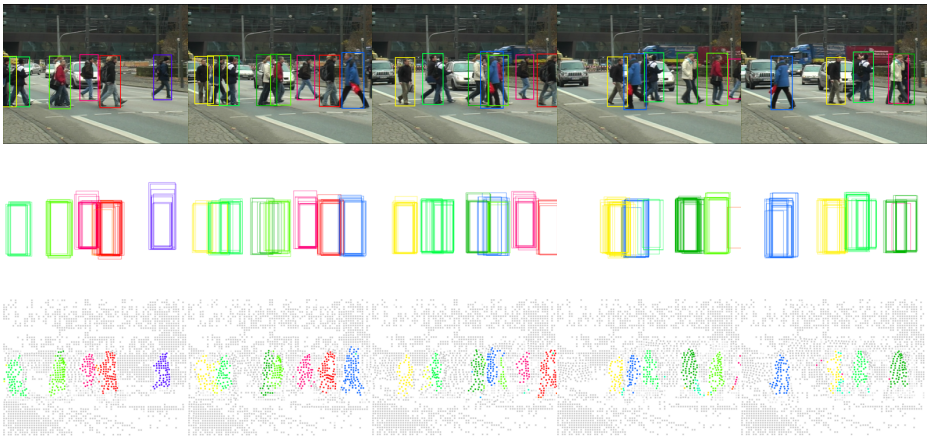


Fig. 7. Results of the proposed Joint Multicut model on the TUD-crossing sequence.

Compared to the state of the art, the proposed method improves the recall on all three datasets. The general tendency is a decrease in the number of false negatives, while the number of false positives is higher than in [28].

	Rcll	Prcsn	FAR	GT	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOTP	MOTAL
TUD-Campus														
Frakiadaki et al.[5]	50.4	57.5	1.89	8	3	2	3	134	178	3	11	12.3	70.1	12.9
Milan et al. [22]	32.6	82.4	0.35	8	1	3	4	25	242	**0**	1	25.6	72.9	25.6
Subgraph MC [28]	83.8	**99.3**	**0.03**	8	5	2	1	**2**	58	**0**	1	83.3	76.9	83.3
Tracklet MC + traj.	**90.3**	94.5	0.27	8	6	1	1	19	**35**	**0**	**0**	85.0	**77.1**	85.0
MCe [14] + det.	82.5	93.7	0.28	8	5	2	1	20	63	3	2	76.0	77.4	76.7
JointMulticut	87.5	98.4	0.07	8	5	2	1	5	45	1	**0**	**85.8**	77.0	**86.0**
TUD-Crossing														
Frakiadaki et al.[5]	75.8	82.3	0.90	13	7	5	1	180	267	13	17	58.3	73.1	59.3
Milan et al. [22]	-	-	0.2	13	3	7	3	37	456	15	16	53.9	72.8	-
Subgraph MC [28]	82.0	**98.8**	**0.05**	13	8	3	2	**11**	198	**1**	**1**	80.9	**78.0**	81.0
Tracklet MC + traj.	85.4	97.7	0.11	13	9	4	0	22	161	5	11	82.9	76.9	83.3
MCe [14] + det.	**92.5**	83.3	1.01	13	12	1	0	204	**83**	14	5	72.7	77.2	73.8
JointMulticut	85.5	97.7	0.11	13	9	4	0	22	160	2	9	**83.3**	77.3	**83.4**
ParkingLot														
Subgraph MC [28]	96.1	**95.4**	**0.45**	14	13	1	0	**113**	95	**5**	18	**91.4**	**77.4**	**91.5**
Tracklet MC + traj.	96.6	93.6	0.66	14	13	1	0	164	85	9	**13**	89.5	76.9	89.9
MCe [14] + det.	**96.8**	88.6	1.23	14	13	1	0	307	**79**	6	15	84.1	77.0	84.3
JointMulticut	96.6	94.9	0.52	14	13	1	0	129	85	6	15	91.1	77.2	91.3

Table 3. Tracking result on multi-target tracking sequences.

A qualitative result is given in Fig. 7 for the TUD-crossing sequence. The bounding boxes overlayed on the image sequence are, for every frame and cluster, the ones with the highest detection score. These were also used for the tracking evaluation. The second row shows all clustered bounding boxes and the third row visualizes the trajectory segmentation. Both detection and trajectory clusters

look very reasonable. Most persons can be tracked and segmented through partial and even complete occlusions. Segmentations provide better localizations for the tracked pedestrians.

4.3 2D MOT 2015

To allow for a comparison to other state-of-the-art multi-target tracking methods, we evaluate our joint multicut approach on the Multiple Object Tracking Benchmark 2D MOT 2015 [50]. In this benchmark, detections for all sequences are provided after Non-Maximum suppression. Thus, the provided detections are already too sparse for the pregrouping into tracklets that has been employed in [28]. As a consequence, we use the detections directly as nodes as it is done for the motion segmentation. The computation of the pairwise terms between detections and between detections and trajectories need adapted as well.

Implementation Details We compute the cut probabilities between detection nodes using Deep Matching [58]. Deep Matching is based on a deep, multi-layer convolutional architecture and performs dense image patch matching. It works particularly well when the displacement between two images is small.

More concretely, each detection d has the following properties: its spatio-temporal location (t_d, x_d, y_d), scale h_d, detection confidence $conf_d$ and a set of matched keypoints M_d inside the detecion d. Given two detection bounding boxes d_1 and d_2, we define $MU = |M_{d_1} \cup M_{d_2}|$, and $MI = |M_{d_1} \cap M_{d_2}|$ between the set M_{d_1} and M_{d_2}. Then the pairwise feature f_e between the two detections no more that 3 frames appart is defined as $(f_1, minConf, f_1 \cdot minConf, f_1^2, minConf^2)$, where $f_1 = MI/MU$ and $minConf$ is the minimum detection score between $conf_{d_1}$ and $conf_{d_2}$.

The computation of pairwise terms between detections and trajectories is performed according to eq. (6) with an undirected template computed as the average of 6 (a) and its horizontally flipped analogon. The sparseness of the detections also alters the statistics of the graph. Assuming that about 20 bounding boxes have been suppressed for every true detection, we weight the links between trajectory and detection nodes by factor 20. We are aware that this is a crude heuristic. Better options would be to learn this factor per sequence type or (better) to use the detections before Non-Maximum suppression which are unfortunately not provided.

Results Our final results on the 2D MOT 2015 benchmark are given in table 4. Compared to the state-of-the-art multi-target tracking method [59], we have an overall improvement in MOTA. Again, we observe a decrease in the number of false negatives while false positives increase.

	FAR	MT	ML	FP	FN	IDs	FM	MOTA	MOTP
Choi [59]	**1.4**	12.2%	44%	**7,762**	32,547	**442**	823	33.7	**71.9**
Milan et al. [22]	**1.4**	5.8%	63.9%	7,890	39,020	697	**737**	22.5	71.7
JointMulticut	1.8	**23.2%**	**39.3%**	10,580	**28,508**	457	969	**35.6**	**71.9**

Table 4. Tracking results on the 2D MOT 2015 benchmark.

5 Conclusion

This paper proposes a Multicut Model that jointly addresses multi-target tracking and motion segmentation so as to leverage the advantages of both. Motion segmentation allows for precise local motion cues and correspondences that support robust multi-target tracking results with high recall. Object detection and tracking allows a more reliable grouping of motion trajectories on the same physical object. Promising experimental results are obtained in both domains with a strong improvement over the state of the art in motion segmentation.

Acknowledgments

M.K. and T.B. acknowledge funding by the ERC Starting Grant VideoLearn.

References

1. Segal, A.V., Reid, I.: Latent data association: Bayesian model selection for multi-target tracking. In: ICCV. (2013)
2. Huang, C., Wu, B., Nevatia, R.: Robust object tracking by hierarchical association of detection responses. In: ECCV. (2008)
3. Wojek, C., Roth, S., Schindler, K., Schiele, B.: Monocular 3d scene modeling and inference: Understanding multi-object traffic scenes. In: ECCV. (2010)
4. Andriluka, M., Roth, S., Schiele, B.: Monocular 3d pose estimation and tracking by detection. In: CVPR. (June 2010)
5. Fragkiadaki, K., Zhang, W., Zhang, G., Shi, J.: Two-granularity tracking: Mediating trajectory and detection graphs for tracking under occlusions. In: ECCV. (2012)
6. Zamir, A.R., Dehghan, A., Shah, M.: GMCP-Tracker: Global multi-object tracking using generalized minimum clique graphs. In: ECCV. (2012)
7. Wojek, C., Walk, S., Roth, S., Schindler, K., Schiele, B.: Monocular visual scene understanding: Understanding multi-object traffic scenes. IEEE TPAMI (2013)
8. Henschel, R., Leal-Taixe, L., Rosenhahn, B.: Efficient multiple people tracking using minimum cost arborescences. In: GCPR. (2014)
9. Tang, S., Andriluka, M., Schiele, B.: Detection and tracking of occluded people. IJCV (2014)
10. Shi, F., Zhou, Z., Xiao, J., Wu, W.: Robust trajectory clustering for motion segmentation. In: ICCV. (2013)
11. Ochs, P., Malik, J., Brox, T.: Segmentation of moving objects by long term video analysis. IEEE TPAMI **36**(6) (Jun 2014) 1187 – 1200
12. Rahmati, H., Dragon, R., Aamo, O.M., Gool, L.V., Adde, L.: Motion segmentation with weak labeling priors. In: GCPR. (2014)
13. Ji, P., Li, H., Salzmann, M., Dai, Y.: Robust motion segmentation with unknown correspondences. In: ECCV. (2014)
14. Keuper, M., Andres, B., Brox, T.: Motion trajectory segmentation via minimum cost multicuts. In: ICCV. (2015)
15. Ochs, P., Brox, T.: Higher order motion models and spectral clustering. In: CVPR. (2012)

16. Andriluka, M., Roth, S., Schiele, B.: People-tracking-by-detection and people-detection-by-tracking. In: CVPR. (2008)
17. Bertasius, G., Shi, J., Torresani, L.: High-for-low and low-for-high: Efficient boundary detection from deep object features and its applications to high-level vision. CoRR **abs/1504.06201** (2015)
18. Fragkiadaki, K., Shi, J.: Detection free tracking: Exploiting motion and topology for segmenting and tracking under entanglement. In: CVPR. (2011)
19. Fragkiadaki, K., Arbelaez, P., Felsen, P., Malik, J.: Learning to segment moving objects in videos. In: CVPR. (2015)
20. Chopra, S., Rao, M.: The partition problem. Mathematical Programming **59**(1–3) (1993) 87–115
21. Deza, M.M., Laurent, M.: Geometry of Cuts and Metrics. Springer (1997)
22. Milan, A., Leal-Taix, L., Schindler, K., Reid, I.: Joint tracking and segmentation of multiple targets. In: CVPR. (2015)
23. Chang, J., Wei, D., III, J.W.F.: A Video Representation Using Temporal Superpixels. In: CVPR. (2013)
24. Andres, B., Kappes, J.H., Beier, T., Köthe, U., Hamprecht, F.A.: Probabilistic image segmentation with closedness constraints. In: ICCV. (2011)
25. Andres, B., Kröger, T., Briggman, K.L., Denk, W., Korogod, N., Knott, G., Köthe, U., Hamprecht, F.A.: Globally optimal closed-surface segmentation for connectomics. In: ECCV. (2012)
26. Kappes, J.H., Speth, M., Andres, B., Reinelt, G., Schnörr, C.: Globally optimal image partitioning by multicuts. In: EMMCVPR. (2011)
27. Keuper, M., Levinkov, E., Bonneel, N., Lavoue, G., Brox, T., Andres, B.: Efficient decomposition of image and mesh graphs by lifted multicuts. In: ICCV. (2015)
28. Tang, S., Andres, B., Andriluka, M., Schiele, B.: Subgraph decomposition for multi-object tracking. In: CVPR. (June 2015)
29. Pirsiavash, H., Ramanan, D., Fowlkes, C.C.: Globally-optimal greedy algorithms for tracking a variable number of objects. In: CVPR. (2011)
30. Wang, X., Turetken, E., Fleuret, F., Fua, P.: Tracking interacting objects optimally using integer programming. In: ECCV. (2014)
31. Shitrit, H.B., Berclaz, J., Fleuret, F., Fua, P.: Tracking multiple people under global appearance constraints. In: ICCV. (2011)
32. Wang, X., Turetken, E., Fleuret, F., Fua, P.: Tracking interacting objects optimally using integer programming. In: ECCV. (2014)
33. Pirsiavash, H., Ramanan, D., Fowlkes, C.C.: Globally-optimal greedy algorithms for tracking a variable number of objects. In: CVPR. (2011)
34. Andriyenko, A., Schindler, K., Roth, S.: Discrete-continuous optimization for multi-target tracking. In: CVPR. (2012)
35. Brox, T., Malik, J.: Object segmentation by long term analysis of point trajectories. In: ECCV. (2010)
36. Lezama, J., Alahari, K., Sivic, J., Laptev, I.: Track to the future: Spatio-temporal video segmentation with long-range motion cues. In: CVPR. (2011)
37. Li, Z., Guo, J., Cheong, L., Zhou, S.: Perspective motion segmentation via collaborative clustering. In: ICCV. (2013)
38. Cheriyadat, A., Radke, R.: Non-negative matrix factorization of partial track data for motion segmentation. In: ICCV. (2009)
39. Dragon, R., Rosenhahn, B.and Ostermann, J.: Multi-scale clustering of frame-to-frame correspondences for motion segmentation. In: ECCV. (2012)
40. Zografos, V., Lenz, R., Ringaby, E., Felsberg, M., Nordberg, K.: Fast segmentation of sparse 3d point trajectories using group theoretical invariants. In: ACCV. (2014)

41. Elhamifar, E., Vidal, R.: Sparse subspace clustering. In: ICCV. (2013)
42. Demaine, E.D., Emanuel, D., Fiat, A., Immorlica, N.: Correlation clustering in general weighted graphs. Theoretical Computer Science **361**(2–3) (2006) 172–187
43. Brox, T., Malik, J.: Large displacement optical flow: descriptor matching in variational motion estimation. IEEE TPAMI **33**(3) (2011) 500–513
44. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. TPAMI **32**(9) (2010) 1627–1645
45. Hoffman, J., Guadarrama, S., Tzeng, E., Hu, R., Donahue, J., Girshick, R., Darrell, T., Saenko, K.: LSDA: Large scale detection through adaptation. In: NIPS. (2014)
46. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: NIPS. (2015)
47. B. W. Kernighan, S.L.: An efficient heuristic procedure for partitioning graphs. Bell Systems Technical Journal **49** (1970) 291–307
48. Beier, T., Kroeger, T., Kappes, J., Kothe, U., Hamprecht, F.: Cut, glue, & cut: A fast, approximate solver for multicut partitioning. In: CVPR. (2014)
49. Beier, T., Hamprecht, F.A., Kappes, J.H.: Fusion moves for correlation clustering. In: CVPR. (2015)
50. Leal-Taix, L., Milan, A., Reid, I., Roth, S., Schindler, K.: Motchallenge 2015: Towards a benchmark for multi-target tracking. arXiv:1504.01942 (2015)
51. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected crfs. In: ICLR. (2015)
52. Papandreou, G., Chen, L.C., Murphy, K., Yuille, A.L.: Weakly- and semi-supervised learning of a dcnn for semantic image segmentation. arxiv:1502.02734 (2015)
53. Uijlings, J.R.R., van de Sande, K.E.A., Gevers, T., Smeulders, A.W.M.: Selective search for object recognition. IJCV **104**(2) (2013) 154–171
54. Girshick, R.: Fast r-cnn. In: ICCV. (2015)
55. E. Horbert, K. Rematas, B.L.: Level-set person segmentation and tracking with multi-region appearance models and top-down shape information. In: ICCV. (2011)
56. Zhao, R., Ouyang, W., Wang, X.: Unsupervised salience learning for person re-identification. In: CVPR. (2013)
57. Cremers, D., Schmidt, F.R., Barthel, F.: Shape priors in variational image segmentation: Convexity, lipschitz continuity and globally optimal solutions. In: CVPR. (2008)
58. Weinzaepfel, P., Revaud, J., Harchaoui, Z., Schmid, C.: DeepFlow: Large displacement optical flow with deep matching. In: ICCV. (2013)
59. Choi, W.: Near-online multi-target tracking with aggregated local flow descriptor. In: ICCV. (2015)