Joint Tracking and Segmentation of Multiple Targets

Anton Milan[1] Laura Leal-Taixé[2] Konrad Schindler[2] Ian Reid[1]
[1]University of Adelaide, Australia [2]Photogrammetry and Remote Sensing Group, ETH Zürich

Abstract

*Tracking-by-detection has proven to be the most successful strategy to address the task of tracking multiple targets in unconstrained scenarios [*e.g. *40, 53, 55]. Traditionally, a set of sparse detections, generated in a preprocessing step, serves as input to a high-level tracker whose goal is to correctly associate these "dots" over time. An obvious shortcoming of this approach is that most information available in image sequences is simply ignored by thresholding weak detection responses and applying non-maximum suppression. We propose a multi-target tracker that exploits low level image information and associates every (super)-pixel to a specific target or classifies it as background. As a result, we obtain a video segmentation in addition to the classical bounding-box representation in unconstrained, real-world videos. Our method shows encouraging results on many standard benchmark sequences and significantly outperforms state-of-the-art tracking-by-detection approaches in crowded scenes with long-term partial occlusions.*

1. Introduction

Despite remarkable progress, automated tracking of multiple targets in unconstrained, crowded environments remains to a large degree unsolved. Noisy and imprecise measurements, long-term occlusions, complicated dynamics and target interactions all contribute to its complexity. Tracking-by-detection has become the method of choice for approaching this problem [40, 55, 56] as it is able to reduce the complexity dramatically by only taking a restricted set of measurements – namely non-maxima suppressed object detections – into account. An obvious downside of this approach is that most of the information available in a video sequence is simply ignored. While state-of-the-art object detectors have reached acceptable performance, both in terms of accuracy [16] and in terms of speed [4, 46], they still consistently fail in cases of occlusion where only a part of the entire object is visible.

We argue that it is beneficial to consider *all* image evidence to handle tracking in crowded scenarios. In contrast to many previous approaches, we aim to assign a unique target ID not only to each individual detection, but to every (super-)pixel in the entire video (*cf*. Fig. 1). This low-level information enables us to recover trajectories of largely occluded targets since the partial image evidence of the superpixels often persists even in the absence of detections.

Figure 1: An example of our instance-based segmentation.

In common with some other approaches [37, 40, 41, 43] we formulate the problem as one of finding a set of continuous trajectory hypotheses that best explains the data, but our approach differs in that we take account of the low-level information in scoring the trajectory hypotheses. We do this by modelling the problem as a multi-label conditional random field (CRF). We show how through judicious and justifiable modelling choices, the CRF energy is submodular and so can be addressed with well-studied optimization techniques such as α-expansion. Our main contributions are:

- a new CRF model that exploits a lot more of the image evidence, including high-level detector responses and low-level superpixel information;
- fully automated segmentation and tracking of an unknown number of targets;
- a complete state representation at every time step that naturally handles occlusions (as opposed to [13, 22]).

Our experimental evaluation on a set of standard, publicly available sequences confirms the advantage of exploiting partial evidence. We are able to improve the recall by 10% on average, while reducing the number of ID switches.

2. Related Work

Multi-target tracking has been and still is an extremely popular research field in computer vision [39]. In this

To appear in the Proc. of the *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, June 2015.

section we will only review some closely related work on tracking-by-detection and segmentation-based methods.

Tracking-by-detection is by far the most explored strategy for multi-target tracking [1, 20, 28, 37, 45, 55, 56]. The main task is split into *(i)* obtaining a set of independent target measurements and *(ii)* resolving the identities of these measurements (performing data association) and connecting them into consistent trajectories. This second part is a lot more challenging than single-target tracking, because the number of possible assignments is exponential in the number of targets and in the number of frames. As a consequence, most formulations aim to either approximate the problem or to find a locally optimal solution.

Early concepts for tracking multiple targets include the multi-hypothesis tracker (MHT) [45] and the joint probabilistic data association (JPDA) [20]. The former builds a hypothesis tree over several frames and aims to find the optimal assignment after heuristic pruning. The latter provides a probabilistic interpretation of all permissible target-to-measurement assignments and relies on gating to keep the problem tractable. More recently, the task has been cast, among others, as integer linear program [5, 28], network flow problem [44, 56], quadratic boolean program [37], continuous or discrete-continuous energy minimization [2, 40], generalized clique graphs [15, 55], and maximum weight-independent set problem [9]. Some can be solved optimally [5, 28, 34, 56], but are rather restrictive in their models, by reducing the target state space only to existing measurements [11, 34, 36, 56], or to a regular lattice [5]. Others are more general but yield complex optimization problems. All of the aforementioned methods have in common that they operate on a set of object detections, which means that all image evidence below a certain likelihood is suppressed and discarded.

To exploit additional image evidence from partially occluded targets, different ideas have appeared in the literature. Izadinia *et al*. [27] make use of the individual part responses of the DPM detector [18] and incorporate these into a "Multiple People Multiple Parts" tracker. Final high-level trajectories are then merged with the network flow formulation [44, 56]. Leibe *et al*. [37] couple the detection task with trajectory estimation to exploit weak detection responses during partial occlusions. Tang *et al*. [50] propose an explicit multi-person detector that can provide detection pairs of side-view pedestrians. That work has also been extended to the more general case of learning occlusion patterns that are a-priori unknown, by analysing tracking failures [49].

Segmentation and Tracking. Semantic image segmentation is the task of assigning every pixel to one particular class label from a predefined set. It is considered an important part of general scene understanding and has been extensively studied in the past [17, 30–32, 47]. Video segmentation [10, 23] extends this idea to video volumes instead of image planes, with the goal of assigning the same label to all pixels that belong to the same semantic object, throughout the entire video sequence.

Video segmentation techniques have also been applied in the realm of multi-target tracking. Bibby and Reid [8] employ the level-set framework to track contours of multiple objects with mutual occlusions, in real time. However, a manual initialization in the first frame is required, which also determines the number of objects. A similar idea, specifically for pedestrian tracking, is also followed in [26, 43]. The method [43] uses a contour tracker to bridge the gaps between sparse HOG detections, and [26] propose a more fine-grained appearance model to better discriminate between foreground and background pixels. The contour representation is however prone to fail when a target becomes occluded.

More recently, Fragkiadaki and Shi [21] have formulated multi-target tracking as clustering of low-level trajectories, so as to enhance tracking in cluttered situations where detectors usually fail. In their following work [22] short, detection-based tracklets are added as high-level cues. Chen *et al*. [13] aim to label supervoxels with their respective target identities (or as background), which is similar in spirit to our work. To that end, they propose a simple greedy optimization scheme based on label propagation with constraints. Again, the segmentation masks in the first frame are marked manually. One limitation of the approaches above is their inherent inability to track targets through full occlusion [13, 22]. In addition, a target's state (*i.e*. its location) is only defined implicitly by the segmentation [13], which makes it rather difficult to estimate the full extent in case of (partial) occlusion. Our proposed method overcomes both limitations by explicitly modelling the continuous state of all targets throughout the entire sequence as a volumetric tube (*cf*. Fig. 2).

3. Approach

Given a video sequence of F frames, our goal is to segment and track all targets within the sequence. More precisely, we seek to estimate the size and location (given by a bounding box) of each target in every frame, and the association of every detection to a target trajectory, and the association of every pixel, either to the background or to one of the target trajectories.

Our high-level approach to this problem is similar to [2, 41]: we generate an overcomplete set of trajectory hypotheses and then optimize an objective that chooses which hypotheses participate in the solution. This objective must capture agreement with image evidence along with our prior beliefs about the properties of valid trajectories such as their continuity, dynamics, *etc*. The details of the trajectory hy-

Table 1: Notation.

Symbol	Description
$\mathcal{V} = D \cup S$	The set of all random variables is a union of superpixel and detections variables.
$\mathcal{E}_S, \mathcal{E}_T, \mathcal{E}_D$	All pairwise edges build spatial, temporal and detection cliques.
ϕ, ψ	Unary and pairwise potentials
ψ^λ	Global potential (label/trajectory cost)
$\mathcal{T}$	Trajectory (4D spline)
$\mathcal{M}, I_\mathcal{M}(\cdot)$	Object shape mask and its intensity

pothesis generation are deferred to Section 6. However, while previous work of this ilk has typically discarded all information other than object detections and modelled trajectories as space-time curves, we propose a more accurate volumetric state representation and a more sophisticated approach to hypothesis evaluation, making use of pixel-level information and aiming to explain *all* of the image data.

We formulate the assignment of detections and (super)-pixels to trajectory hypotheses as a multi-label conditional random field (CRF) with nodes $\mathcal{V} = \mathcal{V}_S \cup \mathcal{V}_D$ and edges $\mathcal{E}$, where $\mathcal{V}_S$ represents all superpixel nodes and $\mathcal{V}_D$ all detection nodes. Each random variable $v \in \mathcal{V}$ can take on a label from the label set $\mathcal{L} = \{1, \ldots, N, \varnothing\}$, which can be either a unique target ID or the background (false alarm) label $\varnothing$. N is the number of trajectory hypotheses, in general much greater than the actual number of targets. Thus, label IDs and trajectory hypotheses are equivalent for our purposes. If a label is assigned to a superpixel or detection that means that the trajectory hypothesis corresponding to that label participates in the solution. A trajectory hypothesis used in the solution is said to be in the active set $\mathcal{T}^*$. $\mathcal{N}$ defines the neighbourhood system on the graph; the edge set $\mathcal{E} = \mathcal{E}_S \cup \mathcal{E}_T \cup \mathcal{E}_D$ includes spatial and temporal relations between neighbouring superpixels, as well as hyper-edges $\mathcal{E}_D$ connecting superpixels with each detection bounding box that encloses it (*cf*. Sec. 4.5).

We aim to find the most probable labelling $\mathbf{v}^*$ for all nodes given the observations, which is equivalent to minimizing the corresponding Gibbs energy: $\mathbf{v}^* = \arg\min_\mathbf{v} E(\mathcal{V})$. We define the energy as follows:

$$E(\mathcal{V}) = \sum_{s \in \mathcal{V}_S} \phi^{\mathcal{V}_S}(s) + \sum_{d \in \mathcal{V}_D} \phi^{\mathcal{V}_D}(d) + \sum_{(v,w) \in \mathcal{E}} \psi(v,w) + \psi^\lambda, \tag{1}$$

with unaries $\phi^{\mathcal{V}_S}$ and $\phi^{\mathcal{V}_D}$ and pairwise potentials ψ.

By involving the pixel (or superpixel) information in the optimization we enable the label IDs to persist even when there is no explicit detector evidence. As we show in the results section (7), this has the effect of boosting recall significantly, especially in scenes exhibiting significant density of targets and occlusion.

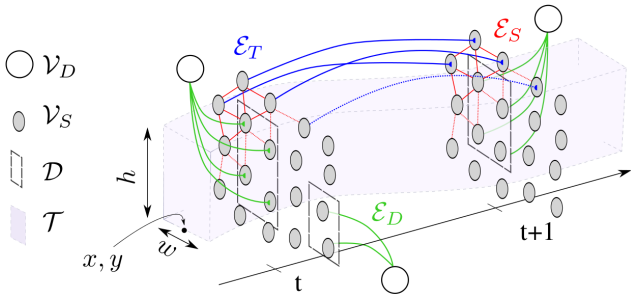


Figure 2: Schematic view of our CRF model for two consecutive frames, showing superpixel nodes $s \in \mathcal{V}_S$, detection nodes $d \in \mathcal{V}_D$ and only a subset of the pairwise edges $\mathcal{E}$ (to keep the figure readable).

An important aspect one has to consider when posing multi-target tracking as a multi-labelling problem is that the number of labels (i.e. the number of distinct trajectories) in the final solution must be inferred during optimization, because targets may enter and exit the field of view. This is in contrast to problems like image segmentation, where the number of semantic classes is typically defined *a priori*. To restrict the number of trajectories from growing arbitrarily high it is therefore necessary to include a regulariser, which favours solutions with fewer labels. We model this by a global factor ψ^λ that also acts as a trajectory-specific prior (see Section 5).

In developing the energy one must take into account the tractability of optimization. In the following section, we will formally define the individual components of the energy E from eq. (1). Our design ensures that the overall energy is submodular and therefore one can apply the standard α-expansion algorithm to find a local minimum efficiently.

4. CRF model

We now discuss the components of the CRF energy in more detail. Our notation is summarized in Tab. 1 and the CRF model (without the label cost) is illustrated in Fig. 2.

4.1. Trajectory model

The motion of each target is modelled as a space-time tube with a rectangular cross-section, represented by a 4D spline $\mathbb{R} \to \mathbb{R}^4$, $\mathcal{T}(t) \mapsto (x, y, w, h)^\top$, with explicit temporal start and end points s and e. Here, x, y are image coordinates of the target's foot position, while w, h are the extents of its bounding box. We choose this very generic representation in order to keep our model flexible, so that it remains applicable to any target class, without requiring a camera calibration or any additional information. We will use $\mathcal{T}(t)_c$ to refer to a specific state component $c \in \{x, y, w, h\}$ at time t.

4.2. Object Shape

Although the bounding box representation of the target (*cf*. Sec. 4.1) keeps the search space manageable, it is a rather crude approximation of the actual shape. For our purpose of segmenting each target, we will therefore use an additional shape prior on multiple occasions within our model. An object-specific foreground mask $\mathcal{M}$ is obtained by averaging multiple annotated silhouettes. In our case, we have used all 2111 annotations of pedestrians from the UrbanStreet dataset [24] to obtain $\mathcal{M}$.

4.3. Observation Model

4.3.1 Target Detections

Like most approaches, we rely on a standard pedestrian detector (we use [14, 51], based on a linear SVM with HOG and HOF features) to obtain a set of putative target locations $\mathcal{D}$. It is well known that state-of-the-art object detectors only work reliably up to a certain degree of occlusion. However, in many real-world scenarios, people occasionally are occluded for long time periods, *e.g*. when walking in a group or when the person density simply gets too high. To track those targets reliably, it is essential to give the tracker access to the very partial information that is still available in the image.

4.3.2 Foreground-background segmentation

To separate foreground (objects) from background, we train a linear SVM online for each (sub)sequence. Note that we do not need any manual supervision but solely rely on the detector output described in the previous section. Positive and negative training samples are obtained by clustering, similar to [48]: The set of negative samples (superpixels) is generated in two ways: *(i)* randomly from all image regions outside detection bounding boxes; *(ii)* explicitly around confident detection boxes to collect a more discriminative set. All negative samples are then clustered into 5 clusters using k-means with mean colour in Lab colour space as feature. Subsequently, all superpixels in the entire sequence are sorted according to the distance to their nearest cluster centre and the last 5%, *i.e*. those farthest from any background cluster, are taken as positive samples (*cf*. Fig. 3).

Finally, a linear SVM is trained using only Lab colour channels as features and each superpixel s_i is assigned a likelihood based on its SVM score:

$$\mathcal{F}^i = \frac{1}{1 + \exp^{\text{-score}}} \tag{2}$$

4.4. Unaries

We define two types of unary potentials, $\phi^{\mathcal{V}_D}$ and $\phi^{\mathcal{V}_S}$ for detection and superpixel nodes, respectively.

4.4.1 Detections

The cost of labeling a node $d_i \in \mathcal{V}_D$ as target j is defined as

$$\phi^{\mathcal{V}_D}(d_i \mapsto j) = w_D \cdot \left(1 - \frac{\mathcal{D}_i \cap \mathcal{T}_j}{\mathcal{D}_i \cup \mathcal{T}_j}\right) \tag{3}$$

and measures the overlap of that detection and any given trajectory hypothesis. The cost of assigning a false positive label to a detection is set to $\phi^{\mathcal{V}_D}(d_i \mapsto \varnothing) = w_\varnothing \cdot d_i^c$, where $d_i^c \in [0, 1]$ is the detection confidence value.

4.4.2 Superpixels

The unary potentials for the set of superpixels $\mathcal{V}_S$ model the likelihood of a superpixel $s_i \in \mathcal{V}_S$ belonging to a particular target. We use colour and optic flow as features and combine both linearly to obtain: $\phi^{\mathcal{V}_S} = \phi^{\mathcal{V}_S}_{\text{col}} + w_{\text{of}} \cdot \phi^{\mathcal{V}_S}_{\text{of}}$.

Colour. The former is defined $\forall j \in \{1, \ldots, N\}$ as

$$\phi^{\mathcal{V}_S}_{\text{col}}(s_i \mapsto j) = \begin{cases} 1 - w_{\mathcal{M}} \cdot \mathcal{F}^i, & \mathcal{S}_i \cap \mathcal{T}_j \neq 0 \\ \infty & \text{otherwise,} \end{cases} \tag{4}$$

where $\mathcal{S}_i$ is the set of all pixels belonging to superpixel s_i. The weight is set to $w_{\mathcal{M}} = I_{\mathcal{M}}(s_i^{\mathbf{x}})$, *i.e*. the intensity of the object prior mask $\mathcal{M}$ at the location of the superpixel s_i, positioned at its tentative location $\mathcal{T}_j$. The cost for labelling s_i as background is defined as $\phi^{\mathcal{V}_S}_{\text{col}}(s_i \mapsto \varnothing) = \mathcal{F}^i$.

Optic Flow. The deviation between the mean optic flow $\bar{s}_i$ of a superpixel s_i and the trajectory j is

$$\phi^{\mathcal{V}_S}_{\text{of}}(s_i \mapsto j) = \begin{cases} w_{\mathcal{M}} \cdot \|\dot{\mathcal{T}}_j - \bar{s}_i\|, & \mathcal{S}_i \cap \mathcal{T}_j \neq 0 \\ \infty & \text{otherwise,} \end{cases} \tag{5}$$

where $\dot{\mathcal{T}}_j$ is the velocity of target j, and $w_{\mathcal{M}}$ the prior mask weight as above.

4.5. Pairwise Edges

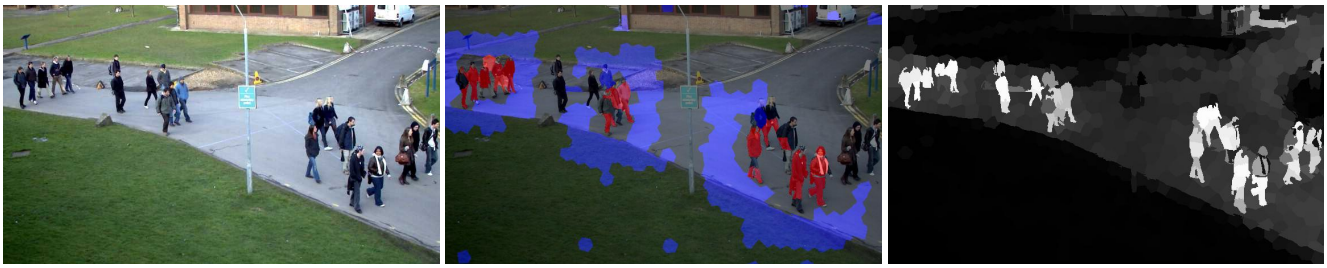
Spatial neighbours. The set of *spatial* edges $\mathcal{E}_S$ consists of all neighbouring superpixels in the image, *i.e*.

$$\mathcal{E}_S = \{(s_i, s_j) | s_i \in \mathcal{N}_S(s_j)\}. \tag{6}$$

In other words, two superpixels are connected if they share an edge in image space.

Temporal neighbours. To obtain the *temporal* neighbourhood $\mathcal{E}_T$ we first run the temporal superpixel segmentation (TSP) method of Chang *et al*. [12] and insert a temporal edge between all superpixels with the same ID in adjacent frames, *i.e*.

$$\mathcal{E}_T = \{(s_i^t, s_j^{t+1}) | \text{TSP}(s_i) = \text{TSP}(s_j)\}. \tag{7}$$



(a) (b) (c)

Figure 3: Foreground / background segmentation as described in Sec. 4.3.2. *(a)* Input frame. *(b)* Positive (red) and negative (blue) superpixel samples collected as training for the SVM. Note that the obtained samples are noisy and contain background regions in the positive set and pedestrians in the negative one. *(c)* Per-superpixel foreground likelihood.

Both for spatial and for temporal connections, the weight of an edge is set according to the mean colour difference:

$$\mathcal{E}^W(s_i, s_j) = \frac{1}{1 + \|\overline{Lab(s_i)} - \overline{Lab(s_j)}\|} \tag{8}$$

High-order cliques. Finally, the edges $\mathcal{E}_D$ connect each superpixel with every detection node that contains it (or none):

$$\mathcal{E}_D = \{(s_i, d_j) | \mathcal{S}_i \cap \mathcal{D}_j \neq \emptyset\}, \tag{9}$$

forming higher-order cliques, closely related to a P^N-Potts potential [29], which can still be solved efficiently. Here, the weight $\mathcal{E}^W$ for each edge is a product of the detection's confidence and the likelihood of a particular superpixel to be labelled as foreground, according to the expected object shape:

$$\mathcal{E}^W(s_i, d_j) = d_j^c \cdot I_{\mathcal{M}}(s_i^{\mathbf{x}}) \; . \tag{10}$$

The high-order cliques enforce consistent labelling within each detection window, which is similar in spirit to the work of Ladický *et al.* [33]. However, our energy includes a parsimony prior modelled as the label cost, which is crucial in the setting with an unknown number of classes.

The potentials for all three edge types take on the same form, to enforce consistent labellings between neighbouring nodes:

$$\psi(s_i, s_j) = w_\psi \cdot [s_i \neq s_j], \tag{11}$$

where $[\cdot]$ is the indicator function.

5. Trajectory cost

Having a complete, global representation of target location and motion, it is straightforward to impose a prior on each trajectory. The total prior cost

$$\psi^\lambda = \sum_{\mathcal{T} \in \mathcal{T}^*} \psi^\lambda_{\text{hgt}} + \psi^\lambda_{\text{ar}} + \psi^\lambda_{\text{dyn}} + \psi^\lambda_{\text{per}} + \psi^\lambda_{\text{lik}} + \psi^\lambda_{\text{reg}} \tag{12}$$

consists of several components outlined below. Note that $\psi^\lambda_{\text{dyn}}, \psi^\lambda_{\text{per}}$, and $\psi^\lambda_{\text{reg}}$ were previously proposed in [2, 41], while $\psi^\lambda_{\text{hgt}}, \psi^\lambda_{\text{ar}}$, and $\psi^\lambda_{\text{lik}}$ are novel and applicable to the volumetric representation with per-pixel likelihood model introduced in this paper.

Height. It reasonable to assume that all targets of a particular class have approximately the same size in 3D space and move on a common ground plane. To prevent unlikely trajectories, whose bounding boxes are either too small or too large, we apply a higher penalty to hypotheses that substantially deviate from the expected target size $\mathcal{H}$. This value is estimated in a data-driven fashion for each sequence, as a pre-processing step. In some cases, the mean size of all highly confident detections may be sufficient. However, to account for perspective distortion, our cost depends on the image location (x, y). In particular, we fit a plane $\mathcal{H} : \mathbb{R}^2 \to \mathbb{R}$ through the 25% strongest responses to obtain an approximate estimate of the target height at every image location. This simple procedure is sufficient for a rough estimation of the target size, without the need for camera calibration. The height prior is then incorporated into the overall trajectory cost as

$$\psi^\lambda_{\text{hgt}}(\mathcal{T}) = w_h \cdot \sum_{t=s}^{e} \left| \mathcal{T}(t)_h - \mathcal{H}\left(\mathcal{T}(t)_x, \mathcal{T}(t)_y\right) \right| . \tag{13}$$

Shape. Following up on the discussion above, targets are also expected to have similar shape. Here, we assume a mean aspect ratio $\rho = \frac{5}{12}$ for pedestrians and penalize the deviation as

$$\psi^\lambda_{\text{ar}}(\mathcal{T}) = w_a \cdot \sum_{t=s}^{e} \left(\mathcal{T}(t)_w / \mathcal{T}(t)_h - \rho \right)^2 . \tag{14}$$

Dynamics. As a motion prior (a.k.a. dynamic model) we

adopt the popular constant velocity model:

$$\psi^{\lambda}_{\text{dyn}}(\mathcal{T}) = w_d \cdot \sum_{t=s}^{e} \left(\tilde{v} - \dot{\mathcal{T}}(t)\right)^2, \tag{15}$$

where $\dot{\mathcal{T}}(t)$ is the target's velocity at time t and $\tilde{v}$ is an offset to penalize deviations from an expected average velocity. We found it beneficial to furthermore apply a small penalty to completely still-standing targets, so as to reduce false positive trajectories on the static background (*cf*. [41]).

Persistence. A persistence cost

$$\psi^{\lambda}_{\text{per}}(\mathcal{T}) = w_p \cdot \left\{ [s > 1] \cdot \mathcal{B}\big(\mathcal{T}(t)\big) + [e < F] \cdot \mathcal{B}\big(\mathcal{T}(t)\big) \right\} \tag{16}$$

is applied to every trajectory that initiates or terminates unexpectedly. $\mathcal{B}\big(\mathcal{T}(t)\big)$ is 1 if the distance to the closest image border exceeds a threshold, and 0 otherwise. This prior reduces track fragmentations and enforces long trajectories.

Image likelihood. To assess whether a trajectory hypothesis correctly explains a target, we exploit the relation of the per-pixel foreground/background likelihood described earlier in Sec. 4.3.2 and the object shape mask $\mathcal{M}$ from Sec. 4.2. Intuitively, the object likelihood at each pixel location should roughly correspond to this mask. Therefore, we define the image likelihood cost for a hypothesis as

$$\psi^{\lambda}_{\text{lik}}(\mathcal{T}) = w_l \cdot \sum_{t=s}^{e} \sum_{i \in S} I_{\mathcal{M}}(s_i^{\mathbf{x}}) \cdot (I_{\mathcal{M}}(s_i^{\mathbf{x}}) - \mathcal{F}^i)^2, \tag{17}$$

where $s_i^{\mathbf{x}}$ is the location of superpixel s_i relative to the hypothesis' bounding box.

Parsimony. Finally, a constant Occam prior $\psi^{\lambda}_{\text{reg}}$ is added to prevent a proliferation of trajectories with too little overall support.

6. Implementation

As noted above, our algorithm relies on an overcomplete set of trajectory hypotheses (or label IDs) from which to construct a solution by minimizing the CRF energy in Eq. (1). The set of initial trajectory hypotheses is generated by *(i)* the fast dynamic programming method of [44]; and *(ii)* a context-free tracker [25], started independently from the most confident detections and run both forward and backward in time.

This initial set is used to begin optimisation of the CRF. As the energy (1) is submodular, one can apply the standard α-expansion algorithm to find a local minimum efficiently. After each α-expansion iteration, the hypothesis space is modified to allow for a broader exploration of the solution space. Similar to [41], active trajectories are extended and shrunk in time; merged to form longer, more plausible trajectories; and newly generated from those detections that are not covered by any hypothesis. For the last option, constant velocity tracklets of four frames ($d_{t-2} : d_{t+1}$) are fitted assuming low velocity, respectively taking into account close-by detections if available to get a more accurate velocity estimate. To ensure that the number of trajectory hypotheses remains manageable, we remove those that have never been picked by the optimizer during the two most recent iterations, and those whose label cost surpasses the total energy. The optimisation terminates when α-expansion cannot find a lower energy, the maximal number of iterations (15) has been reached, or a pre-defined time limit (12 seconds per frame) has been exceeded.

Before presenting experimental results, we also draw attention to a number of implementation factors, to ensure the reproducibility of our method.

Parameters. All parameters are determined automatically by randomly sampling the parameter space as proposed by Bergstra and Bengio [6]. We choose a single parameter set (optimized for MOTA) for all sequences. We found that the amount of occlusion in a scene has a great influence on performance and also on the best choice of parameters. As a proxy for the overall amount of inter-target occlusion, we find the minimum distance between all pairwise detections in each frame and average it across frames to obtain the "mean minimum distance" $\bar{d}$ between detections. This value is then used to modulate the weight of the unary potential for superpixels $\phi^{\mathcal{V}_S}$ in the energy function according to $w_\phi = 1/\bar{d} * w_\phi$; the intuition behind this "occlusion-level specific energy" being that in heavily occluded scenes, where detection is less reliable, it is advantageous to have greater trust in low-level evidence (in our case superpixels), so as to recover largely hidden trajectories.

Pruning. To speed up the optimization, all superpixel nodes that do not intersect with any trajectory hypothesis are discarded from the graph. Note that this does not change the energy because the unaries would force the optimization to label those pixels as background anyway (*cf*. Eqs. (4-5)).

Sliding window. To process video sequences of arbitrary length, we run the optimization over temporal sliding windows of 50 frames at a time, with an overlap of 5 frames. The final result is obtained by bipartite matching of trajectories between adjacent windows, using the Hungarian algorithm.

7. Experiments

7.1. Segmentation

Although the main focus of our paper lies in multi-target tracking, we conduct a small experiment to demonstrate the

Table 2: Quantitative segmentation results on PETS-S2L2.

Method	cl. err.	per-reg. err.	over-seg.	extr. obj
TSP [12]	4.03	29.30	**1.17**	5
Greedy	4.13	25.63	**1.17**	**7**
Ours	**3.56**	**24.34**	1.42	**7**

fidelity of the segmentation that we obtain as a by-product of our optimization. We compare our segmentation to two baselines. TSP is a semi-supervised method that requires a manual initialization in the first frame and then aims to label all superpixels in consecutive frames in a consistent manner [12]. The second one labels all foreground superpixels ($\mathcal{F} \geq .5$) inside bounding boxes obtained by a state-of-the-art tracker [41] in a greedy fashion. To quantify the segmentation performance, we compute four standard error metrics on 5 manually annotated frames of the challenging PETS-S2L2 sequence: the clustering error (percentage of misclassified pixels); the per-region error (average ratio of wrongly labelled pixels per ground truth mask); the number of segments that cover each mask; and the number of extracted objects (those correctly segmented in at least 90% of their area). The quantitative evaluation is summarized in Tab. 2. Video segmentation in this setting turns out to be a very challenging task, even for a human. Our method, although not specifically tuned for this task, is able to outperform the two baselines.

7.2. Tracking

A meaningful quantitative evaluation and comparison of multi-target tracking methods to this day remains a surprisingly difficult task [42]. This is – among other reasons – due to a large and growing number of imprecisely defined evaluation metrics, and even ambiguities in ground truth annotations. We proceed pragmatically and attempt an extensive experimental evaluation and an exhaustive comparison to previously published results, using a collection of the most popular datasets. To facilitate a meaningful comparison for future researchers, we make our code, our data and evaluation scripts as well as the final results publicly available[1]. Additionally, we present our results on *MOTChallenge*, a recent open multi-object tracking benchmark.

Datasets. We have tested our method on seven public sequences totalling over 2,200 frames. Six are part of the widely-used PETS 2010 Benchmark [19], showing a large variability in person count and dynamic behaviour. Five of them contain many long-term occlusions, which make it extremely challenging to detect and track all pedestrians. The last sequence is TUD-Stadtmitte, showing pedestrians in the street filmed from eye level. Segmentation and tracking are challenged by low contrast and similar clothing.

[1] http://research.milanton.net/segtracking/

Table 3: Evaluation in scene space, averaged over six sequences: S2L1, S2L2, S2L3, S1L1-2, S1L2-1, TUDS.

Method	TA	TP	Rcll	Prcn	MT	ML	ID	FM
cl2 [22]	23.1	63.5	41.1	77.1	2	13	50	72
DP [44]	42.1	**65.1**	53.4	91.8	8	11	196	155
DCO [41]	55.7	63.6	61.7	**93.1**	11	9	**49**	**43**
Ours	**58.9**	63.3	**69.2**	88.0	**15**	**6**	54	50

Performance evaluation. Throughout, we use only publicly available detections, ground truth and evaluation scripts [40, 53], to guarantee transparency of the quantitative evaluation.

Further to precision and recall, we report the frequently used CLEAR MOT metrics [7] MOTA and MOTP (abbreviated as TA and TP). The former (tracking accuracy) incorporates the three major error types (missing recall, false alarms and identity switches (ID)) into a single value, such that 100% corresponds to a perfect result with no errors. MOTP (tracking precision) is a measure for the localization error, where 100% again reflects a perfect alignment of the output tracks and the ground truth. Finally, we also quote three popular metrics proposed in [38]. The first two reflect the temporal coverage of true trajectories by the tracker, quantized into three classes: mostly tracked (MT, $> 80\%$ overlap), mostly lost (ML, $< 20\%$), and partially tracked (all others; not shown). The last value we give counts how many times each track is fragmented (FM).

Due to limited space we only show quantitative results averaged over a set of sequences and provide fully detailed tables with further metrics in the supplemental material. The results are quantified in three different settings, found in various publications:

1. Evaluated on the ground plane in scene space, with a 1m threshold (Tab. 3).
2. Same as before but only a rectangular tracking area on the ground is considered (Tab. 4).
3. Evaluated in the image plane with intersection-over-union of bounding boxes as matching criterion (Tab. 5).

The average numbers are shown for six, five and two sequences, reflecting the respective publication. To remain consistent with previously reported numbers we follow the exact same evaluation protocol as all other approaches [3, 40, 41, 52] and use publicly available code [40, 53] for evaluation.

Previous methods. We choose a number of the most recent competitors to directly compare our results to. cl2 [22] is a two-granularity tracking approach that similar to our method leverages both high-level and low-level features to address partial occlusions. However, its inherent inability to

Figure 4: Qualitative tracking and segmentation results on the sequences S2L2 *(top)* and S1L1-2 *(bottom)*.

Table 4: Evaluation in scene space within a pre-defined tracking area, averaged over 6 six sequences *(top)* and 5 sequences *(bottom)*: S2L1, S2L2, S2L3, S1L1-2, S1L1-1.

Method	TA	TP	Rcll	Prcn	MT	ML	ID	FM
CEM [40]	58.8	63.0	66.2	**90.7**	15	10	39	**26**
Ours	**61.6**	**64.2**	**71.6**	88.6	**18**	**7**	**35**	30
H^2T [52]	61.6	62.7	65.2	**95.9**	19	8	**35**	47
Ours	**65.3**	**66.5**	**73.3**	91.2	**22**	**7**	42	**35**

Table 5: Evaluation in image space on S1L1 and S1L2.

Method	TA	TP	Rcll	Prcn	MT	ML	ID	FM
TC [3]	76.6	61.8	**91.9**	85.9	**38**	**1**	**25**	**28**
Ours	**77.7**	**72.2**	88.4	**89.7**	33	3	32	54

track through full occlusions yields inferior tracking results on crowded sequences. Another recent multi-target tracking approach based on superpixels [13] cannot handle small targets present in this data, leading to an even lower performance and is therefore not included in our experiments. Both DCO [41] and CEM [40] formulate multi-target tracking as minimization of a "global" energy, which is similar in spirit to our approach, but only use non-maxima suppressed detections as evidence for the presence of a target. The Hierarchical Hypergraph Tracker (H^2T) [52] builds a hierarchical graph based on detection nodes to assert robust data association between targets close in time, while at the same time bridging long-term occlusions on the upper hierarchy levels. TC [3] addresses the problem by learning an appearance model for each target online and estimating the confidence of each track, to preserve identities through occlusions.

Note that we consistently outperform *all* previous methods in both tracking accuracy (TA) and precision (TP). This is mostly due to the substantial increase in recall, because our method is able to pick up much more image evidence from nearly entirely occluded targets, which allows one to accurately recover more hidden trajectories.

Table 6: Results on the MOTChallenge 2015 Benchmark.

Method	TA	TP	Rcll	Prcn	MT	ML	ID	FM
RMOT [54]	18.6	69.6	40.0	66.4	5.3	53.3	**684**	1282
CEM [40]	19.3	70.7	**43.7**	65.4	**8.5**	**46.5**	813	1023
MotiCon [34]	**23.1**	70.9	41.7	71.1	4.7	52.0	1018	1061
SegTrack	22.5	**71.7**	36.5	**74.0**	5.8	63.9	697	**737**

MOTChallenge. We have also submitted our results to the recent MOTChallenge Benchmark[2] [35]. The 2D MOT 2015 benchmark features 11 test sequences totalling over 5K frames, including moving and static cameras, different frame rates and image resolutions. Tab. 6 lists our results (SegTrack) along with three top public competitors at the time of submission. Our method performs on par with the state of the art but provides per-instance segmentation.

Qualitative results. Fig. 4 shows qualitative results of our method. Both the bounding box representation as well as the pixel-wise segmentation are overlaid on the input frames. Note how we can recover tracks that are largely occluded, as demonstrated *e.g.* by the person ID 12 *(lower left)* or 43 *(upper right)*.

8. Conclusion

We presented a unified CRF model for joint tracking and segmentation of multiple objects in challenging video sequences. Our main innovation is to exploit all image evidence, in particular that of partially occluded objects. Compared to classical approaches based on object detections alone, we are able to significantly improve the number of recovered trajectories in crowded sequences, while at the same time obtaining plausible segmentation masks. In future work we plan to explore how to further improve the segmentation by investigating the influence of additional features.

[2] http://motchallenge.net

Acknowledgements. We gratefully acknowledge the financial support of the Australian Research Council through Laureate Fellowship FL130100102 to IDR.

References

[1] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *CVPR 2008*. 2

[2] A. Andriyenko, K. Schindler, and S. Roth. Discrete-continuous optimization for multi-target tracking. In *CVPR 2012*. 2, 5

[3] S.-H. Bae and K.-J. Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In *CVPR 2014*. 7, 8

[4] R. Benenson, M. Mathias, R. Timofte, and L. V. Gool. Pedestrian detection at 100 frames per second. In *CVPR 2012*. 1

[5] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *IEEE T. Pattern Anal. Mach. Intell.*, 33(9):1806–1819, Sept. 2011. 2

[6] J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13:281–305, Mar. 2012. 6

[7] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: The CLEAR MOT metrics. *Image and Video Processing*, 2008(1):1–10, May 2008. 7

[8] C. Bibby and I. Reid. Real-time tracking of multiple occluding objects using level sets. In *CVPR 2010*. 2

[9] W. Brendel, M. R. Amer, and S. Todorovic. Multiobject tracking as maximum weight independent set. In *CVPR 2011*. 2

[10] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectoriesIn , *ECCV 2010*. 2

[11] A. A. Butt and R. T. Collins. Multi-target tracking by Lagrangian relaxation to min-cost network flow. In *CVPR 2013*. 2

[12] J. Chang, D. Wei, and J. W. Fisher III. A video representation using temporal superpixels. In *CVPR 2013*. 4, 7

[13] S. Chen, A. Fern, and S. Todorovic. Multi-object tracking via constrained sequential labeling. In *CVPR 2014*. 1, 2, 8

[14] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR 2005*. 4

[15] A. Dehghan, S. M. Assari, and M. Shah. GMMCP-tracker: Globally optimal generalized maximum multi clique problem for multiple object tracking. In *CVPR 2015*. 2

[16] P. Dollár, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. *IEEE T. Pattern Anal. Mach. Intell.*, 36(8):1532–1545, 2014. 1

[17] M. Donoser and D. Schmalstieg. Discrete-continuous gradient orientation estimation for faster image segmentation. In *CVPR 2014*. 2

[18] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE T. Pattern Anal. Mach. Intell.*, 32(9):1627–1645, 2010. 2

[19] J. Ferryman and A. Ellis. PETS2010: Dataset and challenge. In *Advanced Video and Signal Based Surveillance (AVSS)*, 2010. 7

[20] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe. Multi-target tracking using joint probabilistic data association. In *19th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, volume 19, Dec. 1980. 2

[21] K. Fragkiadaki and J. Shi. Detection free tracking: Exploiting motion and topology for segmenting and tracking under entanglement. In *CVPR 2011*. 2

[22] K. Fragkiadaki, W. Zhang, G. Zhang, and J. Shi. Two-granularity tracking: Mediating trajectory and detection graphs for tracking under occlusionsIn , *ECCV 2012*, volume 7576. 1, 2, 7

[23] F. Galasso, M. Keuper, T. Brox, and B. Schiele. Spectral graph reduction for efficient image and streaming video segmentation. In *CVPR 2014*, June 2014. 2

[24] H. Gong, J. Sim, M. Likhachev, and J. Shi. Multi-hypothesis motion planning for visual object tracking. In *ICCV 2011*, 2011. 4

[25] J. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *ECCV 2012*, volume 4. 6

[26] E. Horbert, K. Rematas, and B. Leibe. Level-set person segmentation and tracking with multi-region appearance models and top-down shape information. In *ICCV 2011*. 2

[27] H. Izadinia, I. Saleemi, W. Li, and M. Shah. (MP)2T: Multiple people multiple parts tracker. In *ECCV 2012*, volume 6, 2012. 2

[28] H. Jiang, S. Fels, and J. J. Little. A linear programming approach for multiple object tracking. In *CVPR 2007*. 2

[29] P. Kohli, M. Kumar, and P. Torr. P3 beyond: Solving energies with higher order cliques. In *CVPR 2007*. 5

[30] P. Kohli, L. Ladicky, and P. Torr. Robust higher order potentials for enforcing label consistency. In *CVPR 2008*. 2

[31] P. Krähenbühl and V. Koltun. Geodesic object proposalsIn , *ECCV 2014*.

[32] L. Ladicky, C. Russell, P. Kohli, and P. H. S. Torr. Graph cut based inference with co-occurrence statistics. In *ECCV 2010*, volume 5. 2

[33] L. Ladicky, P. Sturgess, K. Alahari, C. Russell, and P. H. S. Torr. What, where and how many? combining object detectors and crfsIn , *ECCV 2010*, Jan. 2010. 5

[34] L. Leal-Taixé, M. Fenzi, A. Kuznetsova, B. Rosenhahn, and S. Savarese. Learning an image-based motion context for multiple people tracking. In *CVPR 2014*. 2, 8

[35] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler. Motchallenge 2015: Towards a benchmark for multi-target tracking. *arXiv:1504.01942*, 2015. 8

[36] L. Leal-Taixé, G. Pons-Moll, and B. Rosenhahn. Branch-and-price global optimization for multi-view multi-object tracking. In *CVPR 2012*. 2

[37] B. Leibe, K. Schindler, and L. Van Gool. Coupled detection and trajectory estimation for multi-object tracking. In *ICCV 2007*. 1, 2

[38] Y. Li, C. Huang, and R. Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *CVPR 2009*. 7

[39] W. Luo, X. Zhao, and T.-K. Kim. Multiple object tracking: A review. *arXiv:1409.7618 [cs]*, Sept. 2014. 1

[40] A. Milan, S. Roth, and K. Schindler. Continuous energy minimization for multitarget tracking. *IEEE T. Pattern Anal. Mach. Intell.*, 36(1):58–72, 2014. 1, 2, 7, 8

[41] A. Milan, K. Schindler, and S. Roth. Detection- and

trajectory-level exclusion in multiple object tracking. In *CVPR 2013*. 1, 2, 5, 6, 7, 8

[42] A. Milan, K. Schindler, and S. Roth. Challenges of ground truth evaluation of multi-target tracking. In *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2013. 7

[43] D. Mitzel, E. Horbert, A. Ess, and B. Leibe. Multi-person tracking with sparse detection and continuous segmentation. In *ECCV 2010*, volume 1. 1, 2

[44] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR 2011*. 2, 6, 7

[45] D. B. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, Dec. 1979. 2

[46] M. A. Sadeghi and D. Forsyth. 30Hz object detection with DPM V5. In *ECCV 2014*. 1

[47] J. Shotton, J. Winn, C. Rother, and A. Criminisi. TextonBoost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV 2006*. 2

[48] G. Shu, A. Dehghan, and M. Shah. Improving an object detector and extracting regions using superpixels. In *CVPR 2013*. 4

[49] S. Tang, M. Andriluka, A. Milan, K. Schindler, S. Roth, and B. Schiele. Learning people detectors for tracking in crowded scenes. In *ICCV 2013*. 2

[50] S. Tang, M. Andriluka, and B. Schiele. Detection and tracking of occluded people. *Int. J. Comput. Vision*, 110(1):58–69, Oct. 2014. 2

[51] S. Walk, N. Majer, K. Schindler, and B. Schiele. New features and insights for pedestrian detection. In *CVPR 2010*. 4

[52] L. Wen, W. Li, J. Yan, Z. Lei, D. Yi, and S. Z. Li. Multiple target tracking based on undirected hierarchical relation hypergraph. In *CVPR 2014*. 7, 8

[53] B. Yang and R. Nevatia. An online learned CRF model for multi-target tracking. In *CVPR 2012*. 1, 7

[54] J. H. Yoon, M.-H. Yang, J. Lim, and K.-J. Yoon. Bayesian multi-object tracking using motion context from multiple objects. In *WACV*, 2015. 8

[55] A. R. Zamir, A. Dehghan, and M. Shah. GMCP-Tracker: Global multi-object tracking using generalized minimum clique graphs. In *ECCV 2012*, volume 2. 1, 2

[56] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *CVPR 2008*. 1, 2