Multi-Target Tracking by Discrete-Continuous Energy Minimization

Anton Milan, *Member, IEEE*, Konrad Schindler, *Senior Member, IEEE*, and Stefan Roth, *Member, IEEE*

Abstract—The task of tracking multiple targets is often addressed with the so-called tracking-by-detection paradigm, where the first step is to obtain a set of target hypotheses for each frame independently. Tracking can then be regarded as solving two separate, but tightly coupled problems. The first is to carry out data association, i.e., to determine the origin of each of the available observations. The second problem is to reconstruct the actual trajectories that describe the spatio-temporal motion pattern of each individual target. The former is inherently a discrete problem, while the latter should intuitively be modeled in continuous space. Having to deal with an unknown number of targets, complex dependencies, and physical constraints, both are challenging tasks on their own and thus most previous work focuses on one of these subproblems. Here, we present a multi-target tracking approach that explicitly models both tasks as minimization of a unified discrete-continuous energy function. Trajectory properties are captured through global label costs, a recent concept from multi-model fitting, which we introduce to tracking. Specifically, label costs describe physical properties of individual tracks, e.g., linear and angular dynamics, or entry and exit points. We further introduce pairwise label costs to describe mutual interactions between targets in order to avoid collisions. By choosing appropriate forms for the individual energy components, powerful discrete optimization techniques can be leveraged to address data association, while the shapes of individual trajectories are updated by gradient-based continuous energy minimization. The proposed method achieves state-of-the-art results on diverse benchmark sequences.

Index Terms—Multi-object tracking, tracking-by-detection, visual surveillance, discrete-continuous optimization

1 Introduction

Multi-target tracking is the problem of simultaneously following multiple moving objects in an image sequence, while also keeping track of individual target identities over time. Weak or incomplete image evidence for each target, occlusions, and complex dynamics all contribute to its difficulty. Following the popular tracking-by-detection paradigm, multi-target tracking consists of two subproblems: Given a video sequence with a set of target observations (or equivalently, detections), the first problem concerns correctly identifying the source of each observation. In other words, each detection needs to be either assigned a unique ID corresponding to a certain target, or identified as a false alarm and therefore discarded from the final solution. The second subproblem consists of inferring the actual trajectories of all targets in the scene in order to obtain their motion patterns over time.

Both aspects of the task are challenging on their own, and much of previous research concentrated on either solving only one [1], [2] or the other [3], [4]. Data association is a combinatorial problem and is typically approached either by simplifying the model, or by settling for an approximate solution (a local minimum of the objective). Approaches that focus on data association [1], [2], [5] typically represent the actual trajectories in an implicit fashion through the associated detections and their locations. This makes it difficult to model target dynamics (cf. [6]), particularly in frames with missing detection evidence, such as during occlusions. On the other hand, estimating target trajectories in their natural continuous space allows one to take into account important trajectory properties such as dynamics, persistence, and collision avoidance. Since data association is dealt with implicitly (e.g., [4]), a very multi-modal energy needs to be minimized over a high-dimensional space, which is rather challenging. Moreover, a unique assignment of detections to targets cannot easily be achieved.

In this paper we aim to unify data association and trajectory estimation in a single model that formulates each aspect in its respective domain through the *minimization of a consistent discrete-continuous energy*. The key advantage of this formulation is that the continuous aspect allows one to model many important trajectory properties that are hard to capture with a purely discrete formulation, while the discrete aspect of the energy makes it possible to leverage powerful discrete optimization techniques for the inherently combinatorial data association. To that end we build on recent advances in multi-model fitting, in particular on the concept of *label costs* [7]. Label costs are incurred only if a particular label is assigned to at least one discrete variable. We show how to formulate multi-target tracking in that framework by incorporating global properties of the trajectory in an associated label cost. Moreover, we introduce a novel *pairwise label cost* to capture pairwise constraints, like physical exclusion at the trajectory level; we extend the

- *A. Milan is with the University of Adelaide, School of Computer Science, Adelaide, SA 5005, Australia. E-mail: anton.milan@adelaide.edu.au.*
- *K. Schindler is with the Photogrammetry and Remote Sensing Group, ETH Zürich, Wolfgang-Pauli-Str. 15, 8093, Zürich, Switzerland. E-mail: konrad.schindler@geod.baug.ethz.ch.*
- *S. Roth is with the Technische Universität Darmstadt, Department of Computer Science, Hochschulstr. 10, 64289, Darmstadt, Germany. E-mail: sroth@cs.tu-darmstadt.de.*

Manuscript received 28 July 2014; revised 28 July 2015; accepted 17 Nov. 2015. Date of publication 2 Dec. 2015; date of current version 12 Sept. 2016. Recommended for acceptance by F. Fleuret.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2015.2505309

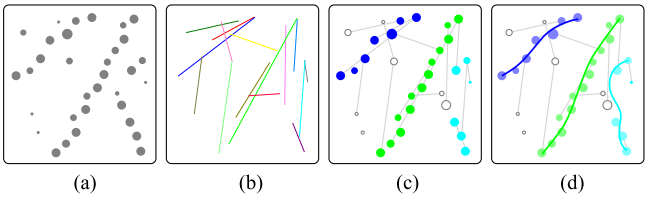

Fig. 1. Given a number of unlabeled object detections (a) and a number of possible trajectory hypotheses (b), our method labels all detections (c) and re-estimates the trajectories (d) using an alternating discrete-continuous optimization scheme.

inference algorithm accordingly. Trajectories are described by cubic B-splines and the fit to a set of detections is modeled in the continuous domain. Data association is modeled by a multi-label Conditional Random Field (CRF) with high-order energy terms, which additionally enforce detection-level exclusion. Two steps are alternated to minimize the resulting discrete-continuous objective, such that trajectory estimation can take advantage of data association and vice versa (cf. Fig. 1). Given a set of trajectory hypotheses, the data association is updated by a variant of the α-expansion algorithm with greedy label removal, taking into account global trajectory properties such as the dynamics and persistence of moving objects through individual label costs. The novel pairwise label cost takes into account the spatial relationship between each pair of trajectories.

In summary, we make the following contributions:

- We formulate a discrete-continuous energy that incorporates both data association and trajectory estimation in a single objective function and models many important aspects of multi-target tracking.
- We enforce target exclusion both at the level of detections and at the level of trajectories to ensure physically plausible solutions.
- We introduce a novel pairwise label cost for discrete multi-label problems that allows one to assign penalties to co-existing label pairs, as opposed to individual labels.
- We describe a suitable minimization algorithm based on α-expansion, greedy label removal, and continuous gradient-based optimization.
- We derive the individual energy components from a statistical analysis of annotated ground truth, to better comply with the empirical statistics of the underlying task.

Parts of this work previously appeared in [8], [9]. Here, we present for the first time the entire formulation including important implementation details, and additional experimental results on new datasets. Moreover, we reformulate the continuous part of the energy to enable a more principled optimization of the entire objective without disregarding the (individual and pairwise) label costs. Our experiments show that minimizing the continuous part using LBFGS combined with QPBO to resolve data association yields best results, and slightly outperforms our previous approach with TRW-S and simplex-based optimization. Finally, we conduct additional experiments, explore various optimization methods, and present a thorough analysis of the individual energy components by examining their influence on the overall performance.

2 Related Work

The body of literature dedicated to multi-target tracking is vast. We will thus review only some of the most important milestones in this field. Early work started several decades ago in the realm of aircraft and naval navigation, where point measurements were obtained with radar and sonar sensors. Perhaps two of the most notable approaches are the multi-hypothesis tracker (MHT) [10] and the joint probabilistic data association filter (JPDAF) [11]. The former boils down to brute-force enumeration with sophisticated pruning heuristics, while the latter follows a probabilistic assignment strategy. Both methods suffer from exponential complexity and are nowadays only rarely used in visual tracking due to their inability to handle crowded real-world scenes.

More recent tracking-by-detection approaches can be roughly categorized into two groups. One encompasses *online* (or *recursive*) state estimation techniques, where the state posterior is inferred based only on the current and past observations. Kalman filters [12] are most widely known for their simplicity, but are (originally) limited to linear models and Gaussian noise distributions. Monte Carlo sampling techniques, such as particle filters [13], [14], [15], are able to overcome these limitations and have thus gained more popularity in the recent past. While non-linear models and multi-modal distributions are typically more faithful representations of the problem, it becomes difficult to maintain the multi-modality with increasing number of targets, since the number of required particles explodes. These filtering methods are usually applied to estimate the target locations (and possibly further continuous state variables relating to their trajectories). A separate procedure, in practice often a greedy assignment or bipartite matching with the Hungarian algorithm, is employed for data association.

One of the main drawbacks of such recursive state estimation is that once an error is made (e.g., due to imprecise or spurious measurements) it cannot be corrected. This is different for the second group of non-recursive tracking approaches [1], [2], [4], [6], [9], [16], [17]. Here, a batch of frames is taken as input, and all trajectories are inferred jointly in light of evidence from all frames within that batch, possibly even the entire sequence.

Interestingly, the bulk of previous work focused on the combinatorial problem of data association, i.e., finding the target (or "origin") that caused each detection. To that end, Jiang et al. [16] proposed a linear objective function with additional constraints to enforce consistency of the spatio-temporal layout between targets. The resulting integer linear program (ILP) is approached by LP-relaxation. Zhang et al. [1] introduced a network flow formulation, where the global optimum of a rather simple objective function can be found in polynomial time with the min-cost flow algorithm. To overcome occlusions, hypothesized detections are inserted heuristically in a post-processing step. Following this line of thought, Pirsiavash et al. [2] presented a much faster greedy algorithm that successively finds the shortest paths in the cost network, and Lenz et al. [18] extended that same framework to an online tracking solution and a memory-bound approximation. Occlusion handling is addressed by Henriques et al. [19], where a track graph with merge/split nodes is constructed and a solution is found with a

flow circulation algorithm. A track graph is also used by Wu et al. [20], where track-linking is converted to a set-cover problem. To include higher-order properties, such as the targets' dynamics, Andriyenko and Schindler [21] and Butt and Collins [6] both suggest to consider frame triplets instead of pairs of neighboring frames. Although such a strategy increases the problem size, it enables one to assess a target's motion locally (i.e., with a unary potential). In the former case the solution is found via LP-relaxation, similar to [16], [17], whereas the latter resorts to Lagrangian relaxation with additional constraints. Further strategies for solving data association include transformation to other graph-theoretical problems like maximum independent set [22] or minimum (multi-)clique [5], [23], and MCMC-type sampling [24], [25].

Berclaz et al. [3], [17], propose to discretize the state space to a regular grid rather than to a discrete set of detections. This allows one to directly optimize for trajectories, with explicit collision avoidance. At the same time, data association is only handled implicitly. The resulting ILP is again solved to (near) global optimality by LP-relaxation [17] or by the k-shortest paths (KSP) algorithm [3]. Their formulation has also been extended to incorporate global appearance modeling in order to better preserve identities over time [26]. In earlier work [4], we have followed a rather opposite strategy, and sidestep the discretization altogether, to obtain an energy over the continuous locations of all targets. The objective is no longer convex, but strong local minima can still be found by a combination of gradient descent and ad-hoc jump moves selected in a greedy fashion.

Here, we present a mixed discrete-continuous model for multiple target tracking. The proposed energy function integrates both tasks at hand, data association and trajectory estimation, while describing each one in its natural domain. Compared to a purely continuous formulation [4] the mixed approach has a number of attractive properties: first, the problem of data association – i.e., measurement-to-target assignment, identifying false measurements and determining the number of targets – is handled within a well-studied graphical model framework, for which well-proven inference algorithms exist. A second aspect is that the data association is modeled explicitly via discrete variables, which is important for a number of applications. Finally, in contrast to previous discrete-continuous approaches based on Markov Chain Monte Carlo (MCMC) sampling [24], [27], the label cost framework makes it rather easy to incorporate arbitrary global trajectory properties into the formulation, and allows one to drop overly restrictive assumptions, such as, e.g., Gaussian motion models.

3 CRF Model

Following the *tracking-by-detection* framework [1], [2], [3], [4], [5], [8], [15], [20], [28], [29], we rely on an independently obtained set of target hypotheses $\mathbf{D}$. To make different tracking systems comparable, we use publicly available detector responses [4], [30], throughout, which are generated by popular object detectors based on HOG [31], HOF [32] and Aggregate Channel Features (ACF) [33]. The individual detections then serve as input data for the reconstruction of trajectories.

The ultimate goal of *tracking-by-detection* is twofold: *(i)* Every detection needs to be explained correctly, i.e., either assigned to a target or identified as a false alarm. Beyond unique assignment of target IDs to detections, it also makes sense to constrain that two simultaneous observations must not be assigned to the same target. *(ii)* The resulting trajectories have to explain the observations in a physically plausible way, i.e., all velocities must remain within physical limits and trajectories must not overlap, because two objects cannot occupy the same physical space at the same time. Especially this latter constraint poses a challenge and has often been neglected in order to keep the optimization tractable [1], [2], [8].

In this work we address both goals in a unified manner. Trajectories are represented by continuous curves, while a discrete multi-label CRF models the data association. Furthermore, the task of mutual exclusion between targets is addressed at two different levels: First, at the level of detections it is handled by introducing pairwise terms between competing detections to avoid an unnatural interpretation of the data. That constraint on its own turns out to be insufficient, however, since it could lead to phantom trajectories that mirror existing ones, but are physically not plausible. We thus also approach exclusion at the trajectory level. We introduce a novel pairwise co-occurrence label cost that is applied only if both labels, i.e., trajectories, are present in a solution. Although both unary and pairwise label costs introduce global factors relating many variables, our proposed optimization scheme is able to (locally) minimize the CRF energy in an efficient and effective manner.

3.1 Notation

Before proceeding to the technical details, let us introduce the notation. To easily distinguish between discrete and continuous variables, a sans-serif font ($\mathsf{a}, \mathsf{b}, \mathsf{A}, \mathsf{B}, \ldots$) is used for discrete variables, while continuous ones are typeset with a standard serif font ($a, b, \ldots$). Bold sans-serif letters ($\mathbf{A}, \mathbf{B}, \ldots$) denote discrete sets, while calligraphic letters ($\mathcal{A}, \mathcal{B}, \ldots$) represent continuous sets. Trajectories are denoted with $\mathcal{T}$. N denotes the number of all trajectory hypotheses and is typically much larger than the number of targets present in a scene. A trajectory $\mathcal{T}_\mathsf{i}$ is represented as a piecewise spline. Trajectory i begins at frame s_i and ends at frame e_i. Its length (in frames) is denoted as $\mathsf{F}(\mathsf{i})$; that of the sequence as F. $\mathbf{O}_{\mathsf{i},\mathsf{j}}$ is the set of frames in which trajectories i and j overlap, and $\zeta^\mathsf{t}_{\mathsf{i},\mathsf{j}}$ their spatial overlap at frame t assuming a target extent e.

D^t_g is the location of one particular detection response in frame t, and $\omega^\mathsf{t}_\mathsf{g}$ its confidence. To refer to the random variable of that detection, we use $\mathsf{d}^\mathsf{t}_\mathsf{g}$. Each random variable $\mathsf{d} \in \mathbf{D}$ is assigned a label $\mathsf{f}_\mathsf{d} \in \mathbf{L}$ from the label set $\mathbf{L} = \{1, \ldots, \mathsf{N}, \varnothing\}$ of all trajectory hypotheses, where $\varnothing$ denotes an outlier label, or equivalently a false alarm. Given a labeling $\mathbf{f}$, we identify "active trajectories" defined as $\mathcal{T}^*_\mathbf{f} = \{\mathcal{T}_\mathsf{i} \,|\, \exists \mathsf{d} \in \mathbf{D} : \mathsf{f}_\mathsf{d} = \mathsf{i}\}$, which include only those trajectory hypotheses associated with detections. The notation is summarized in Table 1.

3.2 Discrete-Continuous Tracking with Label Costs

A large class of labeling problems in computer vision are formulated as MAP estimation, respectively energy minimization, in a discrete, pairwise CRF. This framework also

TABLE 1
Notation

Symbol	Description
F	length of sequence in frames
$\mathcal{T}, \mathsf{N} = \lvert\mathcal{T}\rvert$	set of all N models (trajectory hypotheses)
$\mathcal{T}^*_{\mathbf{f}} \subseteq \mathcal{T}$	set of all active trajectories (given labeling $\mathbf{f}$)
$\mathcal{T}_i$	trajectory i (continuous random variable)
$\mathcal{T}_i(t)$	(x, y)-position of trajectory i in frame t
s_i, e_i	temporal start and end points of trajectory i
$\mathsf{F}(i)$	length of trajectory i in frames ($\mathsf{F}(i) = e_i - s_i + 1$)
$\mathsf{O}_{i,j}$	temporal overlap of trajectories i and j
$\zeta^t_{i,j}$	spatial overlap of trajectories i and j in frame t
e	spatial extent of target
$\mathbf{D}$	set of all detections
d^t_g	detection g in frame t (discrete random variable)
D^t_g	position of detection g in frame t
ω^t_g	confidence of detection g in frame t
$\mathsf{D}(t)$	number of detections in frame t
$\mathbf{L}$	set of all possible labels (trajectory hypotheses)
$f_{d^t_g}$	label of detection g in frame t
$\mathbf{f}$	labeling of all detections
$\varnothing$	outlier label
$\mathbf{E} = \mathbf{E}_{\mathcal{S}} \cup \mathbf{E}_{\mathcal{X}}$	set of all edges of the CRF
$\mathbf{E}_{\mathcal{S}}, \mathbf{E}_{\mathcal{X}}$	temporal smoothness, det.-level exclusion edges
$\Gamma(\cdot, \cdot)$	distance between detection & trajectory
E_{det}	unary potentials of the CRF
$E_{\text{sm}}, E^{\mathsf{X}}_{\text{det}}$	pairwise potentials of the CRF
$E_{\text{traj}}(\mathcal{T}_i)$	(unary) label cost for trajectory i
$E^{\mathsf{X}}_{\text{traj}}(\mathcal{T}_i, \mathcal{T}_j)$	pairwise label cost for trajectories i and j

Each block summarizes the symbols for the continuous part, the detections, the discrete part, and the energy terms, respectively.

serves as the starting point here. To that end, we identify each individual detection $d \in \mathbf{D}$ with a vertex of the graph $\mathbf{G} = \{\mathbf{D}, \mathbf{E}\}$. However, we not only aim to estimate the discrete labeling $\mathbf{f}$, but also the continuous trajectories $\mathcal{T}$. To that end, we propose a unified, discrete-continuous energy that takes the form

$$\begin{aligned} E(\mathcal{T}, \mathbf{f}) = &\sum_{d \in \mathbf{D}} E_{\text{det}}(f_d, \mathcal{T}) \\ &+ \sum_{(d,d') \in \mathbf{E}_{\mathcal{S}}} E_{\text{sm}}(f_d, f_{d'}) + \sum_{(d,d') \in \mathbf{E}_{\mathcal{X}}} E^{\mathsf{X}}_{\text{det}}(f_d, f_{d'}) \\ &+ \sum_{\mathcal{T}_i \in \mathcal{T}^*_{\mathbf{f}}} E_{\text{traj}}(\mathcal{T}_i) + \sum_{\mathcal{T}_i, \mathcal{T}_j \in \mathcal{T}^*_{\mathbf{f}}, i \neq j} E^{\mathsf{X}}_{\text{traj}}(\mathcal{T}_i, \mathcal{T}_j). \end{aligned} \tag{1}$$

3.2.1 The Unary Term

The first term E_{det} models the unary data cost of assigning each detection $d \in \mathbf{D}$ to trajectories by considering their distance. Moreover, it controls which detections are likely to be assigned to the outlier (false positive) class.

3.2.2 Pairwise Terms

The second and third terms E_{sm} and $E^{\mathsf{X}}_{\text{det}}$ are pairwise costs on the detections imposed through two types of pairwise edges $\mathbf{E} = \mathbf{E}_{\mathcal{S}} \cup \mathbf{E}_{\mathcal{X}}$.

Temporal edges $\mathbf{E}_{\mathcal{S}}$ provide temporal smoothing on the data association. All pairs of detections in adjacent frames whose distance is below a threshold τ are connected by an edge (cf. Fig. 2 *left*):

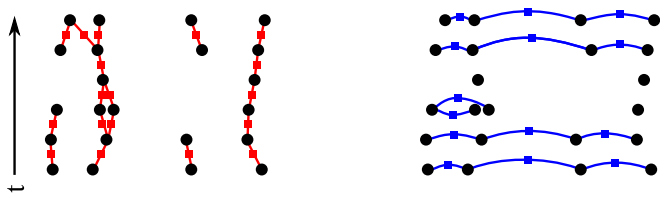

Fig. 2. Factor graph of the underlying CRF with black circular nodes representing the random variables (trajectory hypothesis for each detection) and square nodes representing the pairwise potentials. For clarity, all unary and high-order potentials are omitted. In addition to simple temporal smoothing factors $\mathbf{E}_{\mathcal{S}}$ (*left*, shown in red), we model pairwise exclusion between detections within the same time step $\mathbf{E}_{\mathcal{X}}$ (*right*, blue, subset shown) to prevent implausible data association.

$$\mathbf{E}_{\mathcal{S}} = \left\{ (d^t_j, d^{t+1}_k) \,\middle|\, \left\| D^t_j - D^{t+1}_k \right\| < \tau, t = 1, \ldots, \mathsf{F} - 1 \right\}. \tag{2}$$

The motivation is that nearby detections in adjacent frames should be encouraged to have the same trajectory label. We refrain from longer-range temporal connections, as a large threshold τ would be needed to allow for sufficient target dynamics, coming at the cost of a dense graph and potentially inappropriate label smoothing.

Exclusion edges $\mathbf{E}_{\mathcal{X}}$ ensure a unique target-to-measurement assignment and are motivated as follows. Assuming a target size e, it is impossible that two detections originating from the same frame and being at least the distance e apart are caused by the same object. Therefore, following the same notation as before, we link all simultaneous detections to form

$$\mathbf{E}_{\mathcal{X}} = \left\{ (d^t_i, d^t_j) \,\middle|\, i \neq j, \left\| D^t_i - D^t_j \right\| > e \right\}. \tag{3}$$

For detections that are very close to one another, on the other hand, it is reasonable to accept multiple assignments, since object detectors sometimes erroneously produce multiple outputs from the same object, even after non-maximum suppression. The exclusion factors are illustrated in blue in Fig. 2 *(right)*.

3.2.3 Higher-Order Terms

A-priori knowledge on the individual continuous trajectories, such as their expected dynamics, is captured by the fourth cost term E_{traj}. Because this cost is incurred only for active trajectories $\mathcal{T}^*_{\mathbf{f}}$, i.e., those that are assigned to at least one detection, it takes the form of a label cost [7]. The fifth term provides exclusion at the trajectory level to prevent overlapping trajectories; a cost $E^{\mathsf{X}}_{\text{traj}}$ is incurred when pairs of labels are "active". This term thus constitutes a *pairwise label cost*.

Given the observations $\mathbf{D}$, we aim to estimate the most probable set of trajectories $\hat{\mathcal{T}}$ and labeling $\hat{\mathbf{f}}$ by means of minimizing the corresponding energy: $(\hat{\mathcal{T}}, \hat{\mathbf{f}}) = \arg\min_{\mathcal{T}, \mathbf{f}} E(\mathcal{T}, \mathbf{f})$. We will describe the optimization procedure in detail in Section 4. For now it is important to note that inference will proceed by alternating between continuous optimization of the trajectories $\mathcal{T}$ given a fixed labeling $\mathbf{f}$, and discrete optimization of the data association $\mathbf{f}$ given fixed trajectories $\mathcal{T}$. For continuous optimization we will rely on all terms being differentiable w.r.t. the trajectory parameters. For discrete optimization we exploit move-making algorithms, which allow for tractable approximate inference despite the fact that the label costs E_{traj} and $E^{\mathsf{X}}_{\text{traj}}$ induce

Fig. 3. The trajectory of target i is represented by a two-dimensional piecewise cubic spline with a fixed starting point s_i and a terminating point e_i.

high-order cliques in the corresponding multi-label CRF. But first, let us look at the trajectory representation and the individual energy components in detail.

3.3 Continuous Trajectory Model

In contrast to many existing approaches to multi-target tracking [2], [6], [17], [21], [29], individual trajectories are represented in continuous space. We choose a parametric model, as opposed to an explicit sequence of per-frame coordinates (e.g., [4], [34]). In particular, we use cubic B-splines for that purpose (cf. Fig. 3). This turns out to be a suitable representation for target motion in real-world scenarios, as it avoids discretization artifacts and offers a good trade-off between model flexibility and smoothing of observation noise. More specifically, the two-dimensional spline for each trajectory

$$\mathcal{T}_i : t \in \mathbb{R}_0^+ \rightarrow (x, y)^{\mathrm{T}} \in \mathbb{R}^2, \tag{4}$$

describes the target location $(x, y)^{\mathrm{T}}$ for each point in time t. This two-dimensional representation allows for 2D tracking in image space, or 3D tracking on the ground plane. We assume that the trajectory has a varying number of segments and is parametrized by a coefficient matrix. The number of segments depends on the length of each trajectory and is set to $\max(1, \lfloor \mathsf{F}(i)/10 \rceil)$, where $\lfloor \cdot \rceil$ is the rounding operator. We found that it is advantageous to explicitly model the temporal starting points s_i and end points e_i of each trajectory ($t \in [\mathsf{s}_i, \mathsf{e}_i]$). To ensure that the polynomial does not take on extreme values immediately outside of $[\mathsf{s}_i, \mathsf{e}_i]$, which would prevent other detections in adjacent frames from being assigned to the trajectory later, we add a boundary condition (cf. Eq.(14)) at either end that acts as regularizer by enforcing constant heading.

Assume for now that we are already given a data association $\mathbf{f}$. Then, we can formulate the trajectory estimation problem as minimization of the continuous part of the energy in Eq. (1) given by:

$$E_{\mathbf{f}}^{\mathrm{te}}(\mathcal{T}) = \sum_{\mathsf{d} \in \mathbf{D}} E_{\mathrm{det}}(\mathsf{f}_{\mathsf{d}}, \mathcal{T}) + \sum_{\mathcal{T}_i \in \mathcal{T}_{\mathbf{f}}^*} E_{\mathrm{traj}}(\mathcal{T}_i) + \sum_{\mathcal{T}_i, \mathcal{T}_j \in \mathcal{T}_{\mathbf{f}}^*, i \neq j} E_{\mathrm{traj}}^{\mathsf{X}}(\mathcal{T}_i, \mathcal{T}_j). \tag{5}$$

It follows that $\arg\min_{\mathcal{T}} E_{\mathbf{f}}^{\mathrm{te}}(\mathcal{T}) = \arg\min_{\mathcal{T}} E(\mathcal{T}, \mathbf{f})$ given a fixed labeling $\mathbf{f}$. As mentioned, $E_{\mathrm{traj}}(\mathcal{T}_i)$ and $E_{\mathrm{traj}}^{\mathsf{X}}(\mathcal{T}_i, \mathcal{T}_j)$ are priors on the shape of individual trajectories and on their pairwise relationships, respectively, and will be discussed in detail below.

3.4 Data Cost

For now we focus on the (unary) data cost, which models how well the trajectories $\mathcal{T}$ fit to the hypotheses assigned by $\mathbf{f}$. Assume that we consider the contribution of the g^{th} detection $\mathsf{d} \equiv \mathsf{d}_{\mathsf{g}}^{\mathsf{t}}$ at time t. We define the data cost at time t as:

$$E_{\mathrm{det}}(\mathsf{f}, \mathcal{T}) = \omega_{\mathsf{g}}^{\mathsf{t}} \cdot \begin{cases} \Gamma\big(D_{\mathsf{g}}^{\mathsf{t}}, \mathcal{T}_{\mathsf{f}_{\mathsf{d}}}(\mathsf{t})\big), & \mathsf{f} = \mathsf{f}_{\mathsf{d}} \\ \lambda_{\varnothing}, & \mathsf{f} = \varnothing, \end{cases} \tag{6}$$

where $\omega_{\mathsf{g}}^{\mathsf{t}}$ is the detection confidence, $D_{\mathsf{g}}^{\mathsf{t}}$ is its location and Γ is a distance function. The data cost therefore penalizes the weighted distance to the assigned trajectory. If the detection is labeled as an outlier ($\mathsf{f} = \varnothing$), it is penalized with a constant outlier cost $\lambda_{\varnothing}$, again modulated by $\omega_{\mathsf{g}}^{\mathsf{t}}$. Note that while this outlier cost is irrelevant for minimizing the continuous part of the energy in Eq. (5), it will become important for discrete data association (see Section 3.7).

The motivation behind including the detector's confidence $\omega_{\mathsf{g}}^{\mathsf{t}}$ in the data cost is the following: A low confidence score of the object detector usually means one of two things: either the output is a false alarm, or the bounding box is not properly aligned with the object. It is therefore desirable to impose a lower energy value to weak detections being labeled as false alarms than to strong ones. Analogously, a weak detection should be allowed to be less accurately aligned with the true target, than a strong one.

Regarding the functional form of Γ, it is safe to assume that an object detector will not always localize objects perfectly, but the question remains what pattern the deviations follow. In many cases the potentials (or energy components) are handcrafted, guided by intuition or mathematical convenience. Arguably, it is beneficial to instead derive their functional form from the statistics of the modeled quantities.

3.4.1 Statistical Analysis

To derive a realistic model, we analyze the empirical distribution of the trajectory properties captured in our discrete-continuous energy. We employ eight video sequences (PETS [35] and TUD-Stadtmitte [28]) with ground truth annotations. Note that due to the limited amount of available ground truth data for multi-target tracking, full CRF learning is not the goal here. Instead, we derive a suitable functional form of the potentials. Moreover, it is clear that this comparably small amount of data does not cover all possible tracking scenarios. Thus the goal here is to adapt the tracker to a specific application scenario at hand. Other researchers or practitioners can easily adjust the approach to their specific application case following the same methodology.

Following the relation of model energy and probability (density) as given by the Boltzmann distribution, we study the negative logarithm of the empirical histograms, see Fig. 4. Regarding the data cost, we observe in Fig. 4a that the negative log-probability of the localization error relative to the trajectory grows linearly with the distance. This suggests an absolute value penalty for the data term (thick grey curve in Fig. 4a), respectively an exponential distribution on the distance. In order to enable gradient-based optimization of Eq. (5), we choose the differentiable Charbonnier penalty [36] instead (cf. Fig. 5 *left*):

$$\Gamma(p, q) := \sqrt{\|p - q\|^2 + \epsilon}, \quad \epsilon = 0.1. \tag{7}$$

3.5 Trajectory Prior

In its original form [7], the purpose of a label cost E_{traj} is model selection, i.e., imposing a parsimony prior to keep

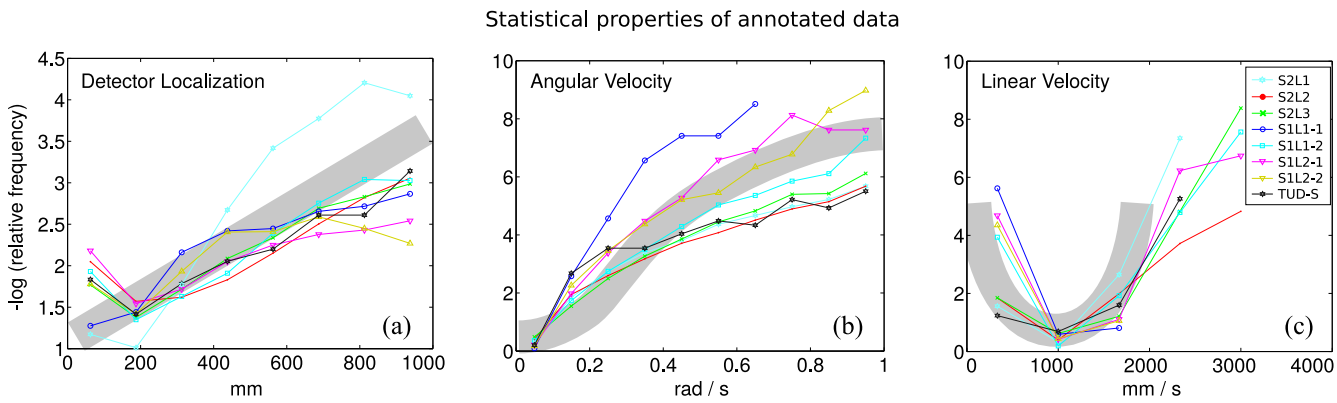

Fig. 4. Empirical analysis of various trajectory properties in multiple people tracking, using ground truth data of eight sequences. Thick grey curves denote our suggested models, motivated by their empirical distributions (negative log-frequency shown).

the number of selected hypotheses (here, trajectories) low. We extend the label cost to also assess the goodness of each individual trajectory. To that end, the cost of trajectory $\mathcal{T}_\mathsf{i}$ being part of the solution is defined as

$$E_{\text{traj}}(\mathcal{T}_\mathsf{i}) = E_{\text{ang}}(\mathcal{T}_\mathsf{i}) + E_{\text{lin}}(\mathcal{T}_\mathsf{i}) + E_{\text{per}}(\mathcal{T}_\mathsf{i}) + E_{\text{bnd}}(\mathcal{T}_\mathsf{i}) + \lambda_{\text{reg}}, \tag{8}$$

which is a combination of terms assessing the angular and linear velocities E_{ang} and E_{lin}, as well as a persistence term E_{per} to reduce the number of short trajectories that start or terminate at unlikely locations. An auxiliary term E_{bnd} includes boundary conditions that prevent the polynomials from attaining extreme values outside their support. Finally, λ_{reg} is a constant penalty term that is added uniformly to all active trajectories and acts as a parsimony prior to favor solutions with fewer targets. To facilitate continuous optimization of the trajectories, all components are modeled as differentiable functions. E_{ang} and E_{lin} are, moreover, derived from a statistical analysis of annotated data (cf. Figs. 4b, 4c). The full trajectory prior is computed as the sum over the individual label costs of all active labels $\mathcal{T}^*_\mathsf{f}$, see Eqs (1) and (5).

3.5.1 Angular Dynamics.

Real objects can only move within physical limits. Our dynamic model assumes that both the angular and the linear velocity change slowly. To formally define the former, let $x \equiv x(t)$ and $y \equiv y(t)$ be the coordinates of a parametric planar curve and $\dot{x}, \dot{y}$ and $\ddot{x}, \ddot{y}$ its first and second temporal derivatives, respectively. The angular velocity at time t is then given as

$$\dot{\theta}(t) = \frac{|\dot{x}(t)\ddot{y}(t) - \dot{y}(t)\ddot{x}(t)|}{\dot{x}(t)^2 + \dot{y}(t)^2}. \tag{9}$$

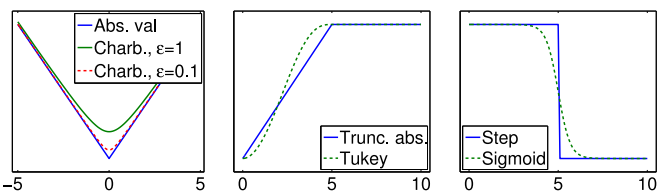

Fig. 5. To enable gradient-based continuous optimization, we choose differentiable penalties *(dotted lines)* approximating the absolute value *(left)*, the truncated absolute value *(middle)*, and the step functions *(right)*.

In order to keep the objective differentiable w.r.t. $\mathcal{T}$, we replace the numerator in Eq. (9) with a suitable approximation, yielding:

$$\dot{\theta}^*(t) = \frac{\sqrt{(\dot{x}(t)\ddot{y}(t) - \dot{y}(t)\ddot{x}(t))^2 + \epsilon}}{\dot{x}(t)^2 + \dot{y}(t)^2 + \epsilon}, \tag{10}$$

with $\epsilon = 0.1$ (cf. Fig. 5 *left*). Note that we also add an ϵ to the denominator to allow unlikely, but potentially possible cases of still standing targets. The empirical distribution of $\dot{\theta}$ shown in Fig. 4b suggests choosing a Lorentzian penalty, respectively Student-t distribution:

$$E_{\text{ang}}(\mathcal{T}_\mathsf{i}) = \lambda_{\text{ang}} \sum_{\mathsf{t}=\mathsf{s}_\mathsf{i}}^{\mathsf{e}_\mathsf{i}} \log\left(1 + \dot{\theta}^*(\mathsf{t})^2\right). \tag{11}$$

3.5.2 Linear Velocity

In addition to the angular velocity we also model the linear velocity. Fig. 4c shows that humans mostly walk at a constant speed of approximately one meter per second. Deviations from that speed are rare such that a quadratic penalty or Gaussian distribution is appropriate:

$$E_{\text{lin}}(\mathcal{T}_\mathsf{i}) = \lambda_{\text{lin}} \sum_{\mathsf{t}=\mathsf{s}_\mathsf{i}}^{\mathsf{e}_\mathsf{i}} \left(\sqrt{\dot{x}(\mathsf{t})^2 + \dot{y}(\mathsf{t})^2 + \epsilon} - 1\text{m/sec.}\right)^2. \tag{12}$$

3.5.3 Persistence

In typical video sequences, objects can appear or disappear only when they enter the field of view (or the tracking area), such that a trajectory will always start and terminate close to a border of a predefined area of interest. Therefore, it is sufficient to define this particular property manually, without an extensive data analysis. To prevent fragmented trajectories and allow a buffer entry zone B, we impose a soft threshold using Tukey's biweight function (cf. Fig. 5 *middle*)

$$E_{\text{per}}(\mathcal{T}_\mathsf{i}) = \lambda_{\text{per}} \prod_{\mathsf{B}} B\left[1 - \left(1 - \left(\frac{\Delta_\mathsf{i}^\mathsf{B}}{B}\right)^2\right)^3\right], \tag{13}$$

where $\Delta_\mathsf{i}^\mathsf{B}$ is the distance of the first, respectively last point of trajectory i to one of the four borders $\mathsf{B} \in \{\text{left}, \text{top}, \text{right}, \text{bottom}\}$. For 2D tracking in image space, these

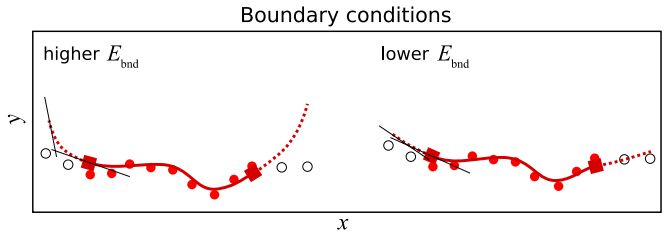

Fig. 6. An additional boundary term E_{bnd} enforces natural trajectory behavior beyond the support. Here, the support, i.e., detections assigned to the red trajectory, are indicated with solid circles. An unlikely configuration with a high (angular) velocity in the dotted area, resulting in a high energy value is shown on the left, while a more plausible trajectory hypothesis is depicted on the right. In the latter case, unassigned detections (empty circles) are more likely to be picked up by data association later.

borders directly correspond to the image borders and $B = 100\,\text{px}$. For tracking in 3D we obtain the tracking area by projecting the image boundaries that delimit the viewing frustum onto the ground plane and set $B = 2\,\text{m}$. Note that we do not impose a hard constraint but rather penalize prematurely terminated trajectories. This allows one to recover tracks that do terminate in the middle of the image (e.g., in the presence of a door or in cases of long-term occlusion).

3.5.4 Boundary Conditions

Fitting a polynomial to a finite set of support points can lead to a solution that attains extremely large values directly outside this support. In our case this would result in unlikely trajectories that quickly accelerate away from their original course, which makes it impossible for neighboring detections to be assigned to them (cf. Fig. 6). To prevent this unnatural behavior, we introduce a boundary penalty

$$E_{\text{bnd}}(\mathcal{T}_{\mathsf{i}}) = \lambda_{\text{bnd}} \cdot \left(\left\| \dot{\mathcal{T}}_{\mathsf{i}}(\mathsf{s}_{\mathsf{i}}) - \dot{\mathcal{T}}_{\mathsf{i}}(\mathsf{s}_{\mathsf{i}} - \Delta) \right\|^2 + \left\| \dot{\mathcal{T}}_{\mathsf{i}}(\mathsf{e}_{\mathsf{i}}) - \dot{\mathcal{T}}_{\mathsf{i}}(\mathsf{e}_{\mathsf{i}} + \Delta) \right\|^2 \right), \tag{14}$$

on either end of the trajectory, to encourage both the velocities and the bearings to be constant for two more frames. We found empirically that setting $\Delta = 2$ yields best results. Since the importance of this term is entirely independent of the data we keep its weight fixed at $\lambda_{\text{bnd}} = 10^{-2}$ in all our experiments.

3.6 Trajectory-Level Exclusion

Let us now turn to the challenging problem of enforcing exclusion at the level of continuous trajectories. It is obvious that multi-target tracking should not allow two or more targets to occupy the same physical space at the same time. However, since the data association is unknown, this seemingly innocuous constraint leads to a hard optimization problem. We will describe our tractable algorithm later in Section 4.

During the discrete optimization step of Eq. (1) the set of candidate trajectories remains fixed. To avoid collisions it is therefore necessary to select only those trajectories (or labels) with no significant spatio-temporal overlap. To that end, we introduce the notion of a *pairwise label cost*. Its purpose is to impose a penalty if two labels co-exist that should not appear simultaneously. In the present case of multi-target tracking such unlikely events occur when two trajectories come too close to each other, causing physically

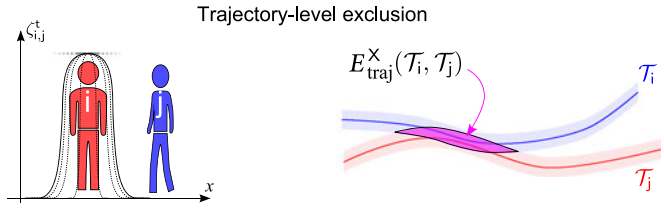

Fig. 7. The distance between two targets is modeled by an isotropic sigmoid *(left)*, while the spatio-temporal overlap between two trajectories is computed by accumulating the overlap at each time step.

implausible situations. For every pair of active trajectories $\mathcal{T}_{\mathsf{i}}, \mathcal{T}_{\mathsf{j}} \in \mathcal{T}^*_{\mathsf{f}}$ with $\mathsf{i} \neq \mathsf{j}$ we thus impose the pairwise label cost $E^{\mathsf{X}}_{\text{traj}}(\mathcal{T}_{\mathsf{i}}, \mathcal{T}_{\mathsf{j}})$. In particular, we choose the co-occurrence penalty to be proportional to the spatio-temporal overlap between two trajectories:

$$E^{\mathsf{X}}_{\text{traj}}(\mathcal{T}_{\mathsf{i}}, \mathcal{T}_{\mathsf{j}}) = \sum_{\mathsf{t} \in \mathbf{O}(\mathcal{T}_{\mathsf{i}}, \mathcal{T}_{\mathsf{j}})} \zeta^{\mathsf{t}}_{\mathsf{i},\mathsf{j}}, \tag{15}$$

which is computed by summing the mutual overlap $\zeta^{\mathsf{t}}_{\mathsf{i},\mathsf{j}}$ over all frames during the common lifespan $\mathbf{O}$ of the trajectories. To keep the label cost differentiable w.r.t. the trajectories, we choose a differentiable, isotropic sigmoidal function (Fig. 5 *right*) around the center of the target:

$$\zeta^{\mathsf{t}}_{\mathsf{i},\mathsf{j}} = \lambda_{E^{\mathsf{X}}_{\text{traj}}} \cdot \left(1 - \frac{1}{1 + \exp(-o_{\mathsf{a}} \| \mathcal{T}_{\mathsf{i}}(\mathsf{t}) - \mathcal{T}_{\mathsf{j}}(\mathsf{t}) \| + o_{\mathsf{b}})} \right); \tag{16}$$

see Fig. 7 for an illustration. The two parameters o_{a} and o_{b} control the size and the falloff of the sigmoid and are directly related to the application-specific shape of the targets. For our experiments we set $o_{\mathsf{a}} = 0.05$ and $o_{\mathsf{b}} = e \cdot o_{\mathsf{a}}/2$, where e is the target extent (diameter).

We point out that the concept of a pairwise co-occurrence label cost is general and not restricted to multi-target tracking. It can trivially be transferred to other applications that involve multi-model fitting, such as semantic segmentation or motion estimation. Note that [37], for example, use a co-occurrence cost to prevent unlikely labeling configurations in the context of semantic segmentation. There, however, the cost is overestimated to keep inference tractable. We prefer to model the cost exactly, but can no longer guarantee global optimality of each expansion step.

3.7 Discrete Data Association

Data association is often the most challenging aspect of tracking multiple targets. We formulate it explicitly as multi-labeling problem, which has the advantage that powerful discrete optimization algorithms can be leveraged. Recalling the notation from above, our goal is to estimate a labeling $\mathbf{f}$ that uniquely assigns each detection $\mathsf{d} \in \mathbf{D}$ to one of the N trajectory hypotheses $\mathcal{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_{\mathsf{N}}\}$, or identifies it as a false alarm using the outlier label $\varnothing$. To that end we minimize the discrete data association energy

$$\begin{aligned} E^{\text{da}}_{\mathcal{T}}(\mathbf{f}) = & \sum_{\mathsf{d} \in \mathbf{D}} E_{\text{det}}(\mathsf{f}_{\mathsf{d}}, \mathcal{T}) \\ & + \sum_{(\mathsf{d},\mathsf{d}') \in \mathbf{E}_{\mathcal{S}}} E_{\text{sm}}(\mathsf{f}_{\mathsf{d}}, \mathsf{f}_{\mathsf{d}'}) + \sum_{(\mathsf{d},\mathsf{d}') \in \mathbf{E}_{\mathcal{X}}} E^{\mathsf{X}}_{\text{det}}(\mathsf{f}_{\mathsf{d}}, \mathsf{f}_{\mathsf{d}'}) \\ & + \sum_{\mathcal{T}_{\mathsf{i}} \in \mathcal{T}^*_{\mathsf{f}}} E_{\text{traj}}(\mathcal{T}_{\mathsf{i}}) + \sum_{\mathcal{T}_{\mathsf{i}}, \mathcal{T}_{\mathsf{j}} \in \mathcal{T}^*_{\mathsf{f}}, \mathsf{i} \neq \mathsf{j}} E^{\mathsf{X}}_{\text{traj}}(\mathcal{T}_{\mathsf{i}}, \mathcal{T}_{\mathsf{j}}), \end{aligned} \tag{17}$$

Fig. 8. Influence of the detection-level exclusion term $E^{\mathsf{X}}_{\text{det}}$, which prevents that two separate targets are assigned the same ID.

which is equivalent to Eq. (1) except that the trajectory hypotheses are kept fixed. The data term E_{det}, as well as the standard (unary) and pairwise label costs $E_{\text{traj}}(\mathcal{T}_{\mathsf{i}})$ and $E^{\mathsf{X}}_{\text{traj}}(\mathcal{T}_{\mathsf{i}}, \mathcal{T}_{\mathsf{j}})$ have been defined above already. We turn our attention to the pairwise terms E_{sm} and $E^{\mathsf{X}}_{\text{det}}$, which model spatio-temporal label smoothing on one hand, and detection-level exclusion on the other.

3.7.1 Temporal Smoothness

Recalling the CRF formulation from Section 3.2 and Fig. 2, the pairwise smoothness terms connect spatio-temporal neighbors $(\mathsf{d}, \mathsf{d}') \in \mathbf{E}_{\mathcal{S}}$ and favor consistent labelings between them with a generalized Potts potential:

$$E_{\text{sm}}(\mathsf{f}_{\mathsf{d}}, \mathsf{f}_{\mathsf{d}'}) = \lambda_{E_{\text{sm}}} \cdot \big(1 - \delta\big[\mathsf{f}_{\mathsf{d}} - \mathsf{f}_{\mathsf{d}'}\big]\big). \tag{18}$$

3.7.2 Detection-Level Exclusion

The exclusion terms are introduced to prevent that two detections within the same time step originate from the same target. The exclusion factor for each edge in $(\mathsf{d}, \mathsf{d}') \in \mathbf{E}_{\mathcal{X}}$ is defined as

$$E^{\mathsf{X}}_{\text{det}}(\mathsf{f}_{\mathsf{d}}, \mathsf{f}_{\mathsf{d}'}) = \begin{cases} \lambda_{E^{\mathsf{X}}_{\text{det}}}, & \mathsf{f}_{\mathsf{d}} = \mathsf{f}_{\mathsf{d}'} \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

The penalty $\lambda_{E^{\mathsf{X}}_{\text{det}}}$ is thus incurred if two distant detections are assigned the same trajectory label (see Fig. 8).

Note that only considering exclusion at the detection level is not enough in order to prevent collisions between targets. In fact the optimization may otherwise be forced to pick two almost identical trajectories (e.g., two trajectories that are close in space and run almost parallel to one another) in order to satisfy these inter-object constraints. It is thus crucial not to disregard the geometric layout of the actual continuous trajectories. We do this through trajectory-level exclusion (Section 3.6).

4 Optimization

Starting from a set of initial trajectory hypotheses and a set of target observations, we aim to find a set of final trajectories that describe the motion pattern of each individual target, and a labeling that explicitly states the origin of each observation. To find a strong local minimum of the energy in Eq. (1), we develop an alternation scheme, which minimizes the energy w.r.t. one set of variables at a time. The optimization stops when the discrete assignment has converged (or a maximum number of iterations has been reached). The complete algorithm is summarized in Algorithm 1. In the following we describe each part in more detail.

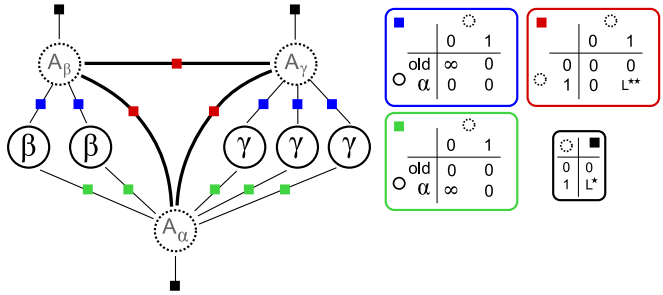

Fig. 9. Factor graph encoding of the unary and pairwise label cost before expanding on α. Random variables (detections) and their current labels (trajectory IDs) are represented by solid circles, while auxiliary variables (higher-order cliques connecting all detections that are part of the same trajectory) are outlined with dashed circles. Solid squares represent unary (black) and pairwise (colored) terms, respectively. The corresponding potentials are depicted on the right with $L^{\star}$ and $L^{\star\star}$ being the respective label cost for a single label (black) and a pair of labels (red). Note that all factors that are unrelated to the label cost are omitted for clarity.

Algorithm 1. Discrete-Continuous Energy Minimization

Input: Initial trajectory hypotheses (Section 5.1), detections $\mathbf{D}$
Output: Labeling $\mathbf{f}$ for $\mathbf{D}$, final trajectories $\mathcal{T}^*_{\mathbf{f}}$
while ¬ converged **do**
  Obtain labeling by minimizing Eq. (1) w.r.t. $\mathbf{f}$
  Greedily remove unnecessary labels (cf. Section 4.1)
  Refit trajectories by minimizing Eq. (1) w.r.t. $\mathcal{T}$
  Modify hypothesis set (cf. Section 5.2)
end
return $\mathbf{f}, \mathcal{T}^*_{\mathbf{f}}$

4.1 Discrete Optimization

Recalling our objective Eq. (1), its discrete part aims to find a labeling $\mathbf{f}$ that assigns a unique trajectory (from a large pool of hypotheses) to each detection. In other words, we want to find a solution to a discrete multi-labeling problem, which amounts to minimizing Eq. (17). This is challenging for several reasons: The pairwise exclusion terms from Eq. (19) have non-submodular costs, the label cost E_{traj} introduces global factors that depend on all labels (cf. [7]), and we need to derive a suitable way to minimize the (non-submodular) pairwise label costs $E^{\mathsf{X}}_{\text{traj}}$. To minimize the label costs, we follow Delong et al. [7] who showed that expansion moves and auxiliary variables can be used to minimize label costs approximately through a series of binary problems with pairwise connectivity.

We will now show how we can also minimize pairwise label costs through expansion moves, similar to [7]. Assume that we already have an intermediate solution $\mathbf{f}$. During a single α-expansion step we now check whether it is beneficial to change any of the labels of $\mathbf{f}$ to α. This is a binary optimization problem [38], where 0 corresponds to no label change and 1 means a variable is switched to label α. Since unary terms E_{det} and pairwise terms $E_{\text{sm}}, E^{\mathsf{X}}_{\text{det}}$ can be simply dealt with as in standard α-expansion [38], we skip the details here and focus on the more challenging unary and pairwise label costs. The corresponding factor graph for a single α-expansion step (without unary and pairwise terms of the original random variables) is depicted in Fig. 9.

Let us first look at the standard per-label cost E_{traj}. Similar to [7], one auxiliary node for each existing label is added (dotted circle) and connected to each variable that carries

the corresponding label. Additionally, an auxiliary node for α, the label to be expanded on, is added and connected to all nodes, regardless of their labeling. However, here we use a different encoding for the auxiliary variables, which act as indicator switches for each label: The auxiliary variable contributes the cost $L^\star$ of having a certain label only once if it is switched on, otherwise its associated cost is 0 (black factors). An infinite pairwise cost prevents the indicator from being off when there is at least one node with the corresponding label (blue and green factors). While this encoding yields supermodular costs (the overall cost is already non-submodular due to Eq. (19)), its purpose will become apparent shortly.

We now turn to the pairwise label cost. Having the same graph structure as above, it is possible to insert a connecting factor between each pair of auxiliary variables (red). To increase the energy value in the case that both labels exist simultaneously, a penalty $L^{\star\star}$ is applied if and only if both corresponding auxiliary variables are switched on.

In each α-expansion step the binary energy is minimized using QPBO [39] rather than standard max-flow, because the energy is not submodular. Note that also message passing algorithms or other inference methods that do not require submodular energies can be used instead (cf. Section 6). We use the OpenGM2 library [40] in our experiments. As it is not guaranteed that each expansion step finds a global minimum of the binary sub-problem, we found it beneficial to add a greedy search step in each expansion move: for each label in turn we check whether the energy can be decreased further by entirely removing that label from the current solution (i.e., replacing the trajectory by the outlier model).

4.2 Continuous Optimization

The continuous part of the proposed energy function is not convex and cannot be minimized in closed form. However, since the energy has been chosen such that its gradient is well defined, any gradient-based optimization algorithm can be employed. We use LBFGS, as it tends to achieve best final performance with comparable run times in our experience. Like with any non-convex optimization, the final result depends on the initialization. We therefore prefer to start from the global minimum of a simplified continuous energy

$$E^*_{\mathsf{f}}(\mathcal{T}) = \sum_{\mathsf{d} \in \mathbf{D}} \widetilde{E_{\text{det}}}(\mathsf{f}_\mathsf{d}, \mathcal{T}), \tag{20}$$

which is equivalent to data cost from Section 3.4, except that $\widetilde{E_{\text{det}}}$ uses the quadratic penalty $\widetilde{\Gamma}(p,q) := \|p-q\|^2$. Eq. (20) is thus a weighted least-squares problem that can be solved in closed form. Note, though, that the global minimum of $E^*_{\mathsf{f}}(\mathcal{T})$ may not necessarily lead to a low energy according to Eq. (1). Hence, the initialization is only used for those trajectories, where the minimizer of the simplified energy also decreases the overall energy value from Eq. (1).

5 Implementation Details

Before presenting our experiments, let us first point out some important aspects regarding the implementation.

5.1 Generating Initial Hypotheses

The optimization is bootstrapped with an initial set of trajectory hypotheses obtained in two ways:

1) Similar to [41], we use a variant of RANSAC to fit straight lines to small randomly chosen subsets of detections (two in our case). To maximize the number of useful trajectory hypotheses, the random sampler prefers detections that are close in space and time. More specifically, two randomly chosen candidate detections $\mathsf{d}_\mathsf{i}^{\mathsf{t}_\mathsf{i}}$ and $\mathsf{d}_\mathsf{j}^{\mathsf{t}_\mathsf{j}}$ are discarded if a linear interpolation between them would result in a target velocity greater than $v = 35\,\text{cm}$ per frame. Otherwise, the acceptance probability is $\exp(-\max(0, |\mathsf{t}_\mathsf{i} - \mathsf{t}_\mathsf{j}| - 4))$. In other words, if the temporal gap between the two candidate detections is four frames or less, and if the linear interpolation results in physically plausible velocity, a new trajectory hypothesis is generated through linear interpolation. If the temporal gap is larger, the acceptance probability is decreased.
2) Additionally, we generate candidate trajectories using two further tracking methods. We employ an extended Kalman filter (EKF) initialized at all detections and using a variety of parameters. Moreover, we use the output of a different multi-object tracker based on dynamic programming [2].

Although different sets of initial trajectory hypotheses may in general lead to slightly different results, we found that the variations of the final solution are marginal.

5.2 Managing the Hypothesis Space

Depending on the initial number of trajectories, a hypothesis space with a fixed number of candidates may be too restrictive to obtain a strong minimum of the energy. To give the optimization more flexibility, we therefore expand the search space after each continuous optimization step, based on the current solution. Note that additional hypotheses do not change the energy function, but only help to better explore the solution space. New hypotheses are generated in a variety of ways:

1) New trajectories are randomly fitted to all detections, as well as specifically to those labeled as outliers. Here we employ the same strategy as described earlier in Section 5.1.
2) Additional trajectories are created by expanding and contracting existing ones in time, as well as splitting them at gaps without detections. Longer trajectories are obtained by simply moving the start or end point (s or e) two frames forward, respectively backward in time (Fig. 10a). Similarly, shorter tracks are added to the hypothesis space by discarding the first, respectively last four frames of existing ones (Fig. 10b). Additionally, existing trajectories that have no support for an extended time period (in our case more than five frames) are used to generate two new shorter tracks that better explain the data by only covering areas with enough detections nearby (Fig. 10c).
3) Two existing trajectories i, j are merged into a new one as long as the temporal gap

$$\Theta = \max(\mathsf{s}_\mathsf{i}, \mathsf{s}_\mathsf{j}) - \min(\mathsf{e}_\mathsf{i}, \mathsf{e}_\mathsf{j})$$

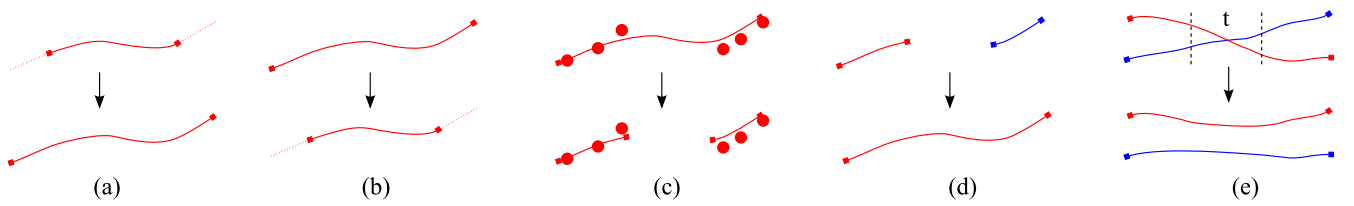

Fig. 10. New trajectory hypotheses *(bottom)* are generated based on existing ones *(top)* by extending *(a)*, shrinking *(b)*, splitting *(c)*, merging *(d)*, and unwinding *(e)*.

is not greater than 20 frames and their combination results in a physically plausible motion, i.e., the target speed does not exceed 1 meter per second (Fig. 10d). Finally, if the shortest distance at time t between two trajectories falls below a threshold, two additional hypotheses are added to avoid intersecting paths. This is achieved by removing a part of each trajectory within the temporal segment $[\mathsf{t}-3,\ldots,\mathsf{t}+3]$ and connecting each end to the other one (Fig. 10e).

Note that in all cases active trajectories are retained to ensure that the energy does not increase. To nonetheless keep the number of possible trajectories from growing arbitrarily, all hypotheses that have a higher label cost than the current value of the overall energy (Eq. (1)) and those that have not been picked up by the discrete optimization within the past two iterations are removed from the hypothesis space.

5.3 Pruning

To speed up inference we prune the graph based on the current hypothesis space before each expansion move. In particular, we remove those random variables from $\mathbf{D}$ whose spatial distance to the trajectory to be expanded upon exceeds $5e$, where e is the target extent, and whose temporal distance to the start, respectively end point of that trajectory exceeds three frames. Note that this also greatly reduces the set of exclusion edges $\mathbf{E}_{\mathcal{X}}$. This does not change the CRF energy in the relevant portion of the solution space (i.e., near a sensible minimum), because the data term already ensures that such detections will never be assigned the same label.

5.4 Parameters

We find a set of eight free model parameters that yield the highest average *MOTA* score by an iterative random search [42]. This strategy is advantageous in practice because it samples the parameter space more efficiently than grid search, but is still largely unsupervised unlike manual search. To reduce the effect of random sampling, we run the optimization with two different random seeds and pick the result with the lowest energy. We state all values used in our experiments in Table 2. The auxiliary parameter $\lambda_{E^*_{\text{det}}}$ that controls the unary weight is included for convenience but can be dropped by dividing all other weights by its value. Note that the seemingly large difference in parameter values between the *PETS/TUD* and the *ETH* datasets arises mainly due to *a)* two completely different setups with a surveillance-type static camera and nearly linear target motion in world coordinates in the first case, and a moving platform with highly non-linear motion patterns in image space in the *ETH* sequences; and *b)* the difference in scale, since the state is computed in millimeters on the ground plane for *PETS/TUD* and in pixels for the moving-camera case. The third dataset, *MOTChallenge*, includes videos with both static and moving cameras, large variation in target size, different motion patterns, etc. Hence, a mixture of both parameter sets shows best performance.

5.5 Sliding Window

Even though the presented method can be applied to entire sequences, we found it beneficial, both in terms of speed and in terms of accuracy, to perform the optimization over smaller temporal windows (cf. Fig. 11). It may seem surprising that a sequence of local solutions yields better performance. This can be explained by the fact that the optimization is not able to explore the vast solution space of an entire sequence within a reasonable time frame. The length of each window is set to 50 frames and successive windows overlap by five frames, however, these values have not been optimized. To ensure seamless correspondence between adjacent windows, trajectories with a significant spatio-temporal overlap within the overlapping time interval are merged.

6 Experiments

We present a thorough quantitative evaluation of our proposed approach, as well as a comparison to a range of previous methods.

Moreover, our framework cleanly separates modeling from inference, hence we also conduct an analysis of various optimization algorithms for both the discrete and the continuous part of the model. Finally, we investigate the importance and influence of individual energy terms by applying different weightings.

6.1 Optimization Strategies and Runtime

To minimize the energy in Eq. (1), one needs on the one hand a discrete inference algorithm for the labeling $\mathbf{f}$, and on the other hand a continuous optimization procedure to find the trajectories $\mathcal{T}$. We have conducted experiments to investigate the performance of various algorithms including tree-reweighted sequential message passing (TRW-S) [43], min-sum loopy belief propagation (LBP) [44], [45], and quadratic pseudo-Boolean optimization (QPBO) [39]; as well as non-

TABLE 2
Parameters Used in Our Experiments

Dataset	$\lambda_{E^*_{\text{det}}}$	λ_{reg}	λ_{ang}	λ_{lin}	λ_{per}	$\lambda_{E^{\mathsf{X}}_{\text{traj}}}$	$\lambda_{\varnothing}$	$\lambda_{E_{\text{sm}}}$	$\lambda_{E^{\mathsf{X}}_{\text{det}}}$
PETS/TUD	320.3	0.3	2e-4	13.3	22.1	29.6	520.1	0.2	19.5
ETH	15.5	592.7	250.8	37.4	0.1	186.9	299.8	0.02	173.2
MOTCh.	14.1	777.1	5e-5	1e-5	5e-4	394.5	298.7	0.2	37.1

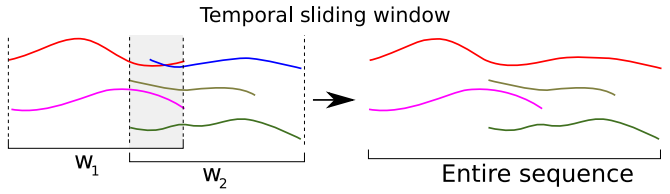

Fig. 11. The optimization is performed independently for each batch of frames *(left)*. Overlapping trajectories (red and blue) are then merged to obtain the final result *(right)*.

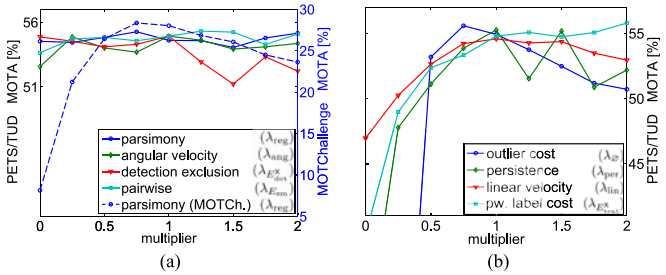

Fig. 12. Influence of different energy terms on the overall tracking performance. Each curve indicates the change of performance measured by MOTA if one individual parameter is altered while keeping all the others fixed. Note that the nominal value (multiplier = 1) does not always yield the highest *MOTA* score, since the parameter set has been determined through a stochastic search and may thus not be globally optimal for the given data.

linear conjugate gradient descent (CGD), limited-memory BFGS (LBFGS), and the derivative-free Nelder-Mead (or simplex) scheme. The results are listed in Table 3. The combination of QPBO and LBFGS yields best performance w.r.t. to *MOTA* and is thus used throughout the remainder of the paper.

Our Matlab code takes $\sim 0.5\,\mathrm{s}$ per frame on average to converge (excluding the detector). With an optimized implementation real-time performance is within reach.

6.2 Energy Analysis

To gain a better understanding of the role of the individual energy components and how they contribute to the entire model, we perform a more in-depth analysis. To that end, we vary the value of each parameter in the range $[0, 2]\times$ of its nominal value and observe the change in performance as measured by the tracking accuracy *MOTA* (cf. Fig. 12). We show the influence of each individual term, while holding all the other ones fixed. The parameters are separated into two groups.

Fig. 12a shows that at the coarse level of overall *MOTA*, the final result is robust to the values of $\lambda_{\mathrm{reg}}, \lambda_{\mathrm{ang}}, \lambda_{E^{\mathsf{x}}_{\mathrm{det}}}, \lambda_{E_{\mathrm{sm}}}$ and exhibits only small variations of the tracking accuracy. It is important to note, though, that the importance of a parameter can vary between different sequences and/or settings. E.g., the parsimony term λ_{reg} may not have much influence on *PETS/TUD*, but is very significant for other datasets, such as the *MOTChallenge* training set. Moreover, this parameter gives the user intuitive control over the precision/recall trade-off, an important tuning decision for practical systems that is not well captured by the compound *MOTA* metric.

A further limitation of the scalar *MOTA* is that it does not reflect the influence of the detection-level exclusion penalty $\lambda_{E^{\mathsf{x}}_{\mathrm{det}}}$. Dropping that term yields visually prominent errors especially for people walking close to each other (cf. Fig. 8), whereas averaged *MOTA* is dominated by false positive trajectories and long-term occlusions. The weights λ_{ang} and $\lambda_{E_{\mathrm{sm}}}$ for angular velocity and label smoothing, respectively, also show similar behavior over a range of values, but dropping them completely decreases the *MOTA* by $\approx 5\%$. It thus appears sensible to include them despite the somewhat increased model complexity.

The second group $(\lambda_{\varnothing}, \lambda_{\mathrm{per}}, \lambda_{\mathrm{lin}}, \lambda_{E^{\mathsf{x}}_{\mathrm{traj}}})$ in Fig. 12b shows a larger influence on the performance when the respective energy term is downweighted. In particular, it is essential to include the outlier cost $\lambda_{\varnothing}$, the pairwise label cost $\lambda_{E^{\mathsf{x}}_{\mathrm{traj}}}$, and the persistence term λ_{per}. The former penalizes trivial solutions with no targets. The pairwise label cost prevents competing trajectories to "share" the detections from the same target, thereby reducing the number of false tracks. The persistence penalty enforces longer, uninterrupted trajectories, reducing the number of ID switches.

TABLE 3
Performance Comparison of Different Optimization Techniques w.r.t. Ground Truth

Method	MOTA	MOTP	ID	final E	converged
QPBO+ LBFGS	56.3%	66.0%	42	45,417	114 s
QPBO+ NM	55.9%	63.5%	50	45,408	109 s
QPBO + CGD	55.3%	63.5%	60	45,173	100 s
LBP+ LBFGS	54.9%	63.2%	61	45,805	131 s
TRWS + LBFGS	54.5%	63.4%	57	45,774	127 s

Energy values shown are averaged over five runs on the first 50 frames of PETS-S2L2.

6.3 Datasets and Metrics

We evaluate our tracker on 19 video sequences. The first set consists of six sequences for which camera calibration is available, which enables tracking in world coordinates on the ground plane. Besides the widely used *PETS S2.L1* sequence, we also include four more challenging scenarios from the same dataset: *S2.L2*, *S2.L3*, *S1.L1-2* and *S1.L2-1*. The *PETS* benchmark [35] shows pedestrians walking across an intersection in various directions at variable speed. The number of people varies from a few to as many as 40. We demonstrate the applicability of our method on *monocular* video sequences, and therefore only use the first camera view point (out of eight recorded) in all experiments. Multi-camera detection and tracking is out of the scope of this paper. *TUD-Stadtmitte* offers a different setup. Here, a busy pedestrian street is filmed from a low camera angle. In both cases, tracking is performed on a ground plane obtained from camera calibration. As a second set, we employ the widely used *BAHNHOF* and *SUNNY DAY* sequences from the ETH Mobile Scene *(ETHMS)* dataset [46], where a busy pedestrian street is filmed from a moving stereo camera. Note that we do not use the available camera calibration and depth maps for these sequences, but rather track people in image space. To account for strong perspective distortion, we scale the targets' linear and angular velocities with $1/\mathsf{Ch}(\mathcal{H}_y - y_{\mathsf{i}})$ along the y-coordinate of target i, where Ch is the Charbonnier approximation of the absolute value as before and $\mathcal{H}_y$ is the estimated horizon location in the image, which is set to 250 for these two

TABLE 4
Quantitative Results on Each Test Sequence Measured in 3D World Coordinates

Sequence	MOTA	MOTP	FP	FN	ID	MT	ML	FM
S2.L1	87.4%	76.1%	371	188	25	18	0	12
S2.L2	57.0%	59.8%	485	3,801	137	10	3	131
S2.L3	40.1%	65.1%	74	2,527	22	9	23	21
S1.L1-2	58.7%	69.4%	150	1,427	12	19	14	8
S1.L2-1	29.8%	60.1%	104	3,400	47	5	23	38
Stadtmitte	64.5%	65.3%	98	303	9	5	0	11
mean	56.3%	66.0%	214	1,941	42	11	11	37

sequences. Finally, we also provide quantitative results on the recent *MOTChallenge* benchmark[1] [30] that incorporates 11 training and 11 test sequences including various person sizes, camera motion, image resolutions, etc., and provides a centralized evaluation server to ensure fair and meaningful comparisons.

We report the widely accepted *CLEAR MOT* [47] metrics evaluated in 3D with a 1 m hit/miss threshold. The Multiple Object Tracking Accuracy *(MOTA)* combines all false positives *(FP)*, false negatives *(FN)*, and identity switches *(ID)* into a single number, and the Multiple Object Tracking Precision *(MOTP)* measures the average distance between the ground truth and the tracker output. Note that both metrics are normalized such that 100 percent corresponds to no errors and perfect alignment, respectively. To better assess the quality, we additionally report the numbers of Mostly Tracked (MT, ground truth trajectory covered by a track for $\geq 80\%$ of its time span) and Mostly Lost (ML, $\leq 20\%$) trajectories. Finally, track fragmentation *(FM)* counts how often a ground truth trajectory is not covered by any hypothesis [48]. Table 6 also shows the numbers for recall and precision. These figures are produced with a 2D evaluation protocol using a publicly available implementation.[2]

6.4 Quantitative and Qualitative Evaluation

Table 4 lists the quantitative performance of our discrete-continuous optimization on each of the six test sequences with calibrated cameras, as well as the average performance.[3] A comparison to other methods is given in Table 5. The results obtained by an extended Kalman filter with greedy data association serve as baseline. Additionally, we report the results obtained by five publicly available trackers: the multi-hypothesis tracker [10], the boosted particle filter (BPF) [14], the grid-based approach based on the k-shortest paths algorithm (KSP) [3], a network-flow formulation solved via dynamic programming (DP) [2], and a continuous energy minimization approach [4]. We used the original implementations provided by the authors. For the classical MHT, for which no original code is available we use the implementation by David Antunes.[4] Note that CEM [4] and KSP [3] compute the tracking results on a predefined rectangular tracking area and must therefore be compared separately from the other techniques.

TABLE 5
Quantitative Comparison of Our Method to Previous Work, Averaged over Six Sequences: Full Field of View (*top*) and Cropped Tracking Area (*bottom*).

Method	MOTA	MOTP	FP	FN	ID	MT	ML	FM
Det. [31], [32]	–	–	743	1,928	–	–	–	–
ours	56.3%	66.0%	214	1,941	42	11	11	37
MHT [10]	45.8%	63.2%	485	2,036	120	9	9	137
DP [2]	45.4%	65.1%	117	2,543	154	8	11	132
EKF	36.6%	62.5%	87	2,407	24	3	20	42
BPF [14]	33.8%	61.4%	602	2,638	103	6	13	181
ours	58.3%	67.0%	163	1,424	29	15	12	23
CEM [4]	57.0%	64.2%	152	1,509	43	13	10	29
KSP [3]	52.2%	60.4%	264	1,368	46	15	9	65

The first row of Table 5 shows the recall and precision of the detections used as input for all trackers. Note that various approaches may achieve their highest performance using different detection methods, which makes it nearly impossible to only compare the tracking component of each algorithm. KSP, for instance, expects a dense likelihood map on the entire grid. Therefore, we smooth each detection with a Gaussian, thus allowing it to contribute to multiple neighboring cells. The width of the Gaussian is chosen empirically to achieve best performance. Instead of these Gaussian "soft detections", we have also experimented with using the raw detector likelihood at each grid cell. Unfortunately, in our setting with monocular video and strong perspective distortion, the detection maps become too diffuse, which leads to false positives. With this setup we were unable to outperform the smoothed discrete detection maps, which we thus used in the following. All other methods (MHT, BPF, CEM and DP) expect discrete bounding boxes as input in their original implementations. We attempt to keep the comparison of individual trackers as fair as possible and therefore use the exact same set of detections and ground truth for each one. Note that we use generic, off-the-shelf pedestrian detectors ([32] for *PETS/TUD* and *ETH*, [33] for *MOTChallenge*), not specifically customized for a particular tracker. Moreover, we tune the parameters of each approach for best accuracy using a variant of the iterative random search [42]. The same set of parameters is used per scenario: one fixed parameter set for the six sequences that are tracked in 3D world coordinates, one set for the two *ETH* videos filmed from moving cameras, and one set for the *MOTChallenge* benchmark (cf. Table 2).

Our proposed method shows clear advantages in crowded scenarios, producing the best *MOTA*. Averaged over all sequences, the result is 10 percentage points higher than MHT or the recent dynamic programming approach (DP) and over 20 percentage points higher than the boosted particle filter. It is worth noting that the localization error (MOTP) hardly changes between different methods. Even though our continuous state representation is in principle able to correct inaccurate location estimates of the detector, it does not greatly improve localization performance.

1. www.motchallenge.net
2. iris.usc.edu/people/yangbo/downloads.html
3. Please note that the sequences *S2L2* and *S1L2-1* were incorrectly evaluated in our previous paper [9] using a cropped ground truth. The corrected numbers are stated on the project website at research.milanton.net/dctracking/.
4. www.multiplehypothesis.com/

Fig. 13. Exemplar frames (slightly cropped) from the sequences *TUD-Stadtmitte* [28], PETS *S2L2* [35], and *ETH Bahnhof* [46].

However, one should keep in mind that MOTP is highly sensitive to the annotations, and that evaluation artifacts may arise, e.g., due to non-optimal ground-truth-to-tracker assignments or consistently inaccurate bounding boxes.

Fig. 13 shows selected frames of the tracker output from three sequences. Note that even in crowded sequences like *S2L2*, our energy minimization approach is able to correctly localize and identify most targets, including those that are partially or fully occluded, while successfully suppressing false detections.

We also evaluate our method on two sequences from the *ETHMS* dataset [46], see Table 6. We use the detector output from [29], [49], and the publicly available evaluation script. State-of-the-art methods for these sequences heavily rely on tracklet linking through significant periods of occlusion, based on appearance and other cues. Since our CRF does not model these, we post-process our tracker output with a simple extrapolation-based track linking scheme (please refer to Appendix A for a detailed description). While our simplistic linking scheme leads to comparatively many ID switches, the high recall and precision numbers indicate that our discrete-continuous CRF yields a competitive basis for appearance-based occlusion handling.

TABLE 6
Quantitative Comparison to Three State-of-the-Art Methods on the *ETHMS* Dataset [46]

Method	Recall	Precision	MT	ML	FM	ID
DP [2]	67.4%	91.4%	50.2%	9.9%	143	4
PIRMPT [49]	76.8%	86.6%	58.4%	8.0%	23	11
Online CRF [29]	79.0%	90.4%	68.0%	7.2%	19	11
Our Method	76.2%	87.6%	58.3%	7.1%	78	43

Finally, Table 7 lists our results on the *MOTChallenge 2015* benchmark along with six selected submissions. This table shows three additional measures: The average rank (ARk) over all shown metrics, the variance in *MOTA* across all test sequences, as well as the relative ID switches, which are computed as ID/Recall. The low number of both absolute and relative identity switches once again underlines the power of graphical models and discrete inference for data association. Our method is also fairly robust across all scenarios, as indicated by a quite low variance of the tracking accuracy.

We make our source code and other data, including detections and ground truth, publicly available.[5]

7 Conclusion and Future Work

We have presented a mixed discrete-continuous formulation of multiple target tracking. Discrete data association and continuous trajectory estimation are integrated in a single, unified energy function over all targets. The proposed energy captures many important constraints, both between different observations of the same target and between different targets moving in the same physical space. As far as possible the functional form of the constraints is derived by a statistical analysis of real-world data. A novel pairwise label cost for multi-label CRFs enables accurate exclusion handling at the level of continuous trajectories. In conjunction with the model we have also designed an efficient

5. http://research.milanton.net/dctracking/

TABLE 7
Quantitative Results on 2015 2D MOTChallenge [30]

Method	ARk	MOTA	Var	TP	ID	rel. ID	FM
ours	2.0	19.6	14.1	71.4	521	13.8	819
SegTrack [50]	2.5	22.5	15.2	71.7	697	19.7	737
MotiCon [51]	4.0	23.1	16.4	70.9	1,018	24.4	1,061
CEM [4]	4.3	19.3	17.5	70.7	813	18.6	1,023
RMOT [52]	4.7	18.6	17.5	69.6	684	17.1	1,282
SMOT [53]	4.7	18.2	10.3	71.2	1,148	33.4	2,132
DP [2]	5.8	14.5	13.9	70.8	4,537	104.7	3,090

algorithm to minimize the resulting non-convex and non-submodular energy function. Experimental results show a clear benefit of the proposed approach. Despite the fact that it can only be optimized locally, the integrated discrete-continuous energy formulation outperforms several existing methods on standard benchmark sequences.

An interesting extension of the proposed model would be to integrate target appearance as an additional cue to improve robustness, especially against identity switches. It also appears feasible, at least conceptually, to extend the graphical model so as to bridge long-term occlusion gaps to further reduce the number of missed targets.

APPENDIX

Track linking. In this section we briefly outline our track linking scheme from Section 6.4. Starting from a final set of T tracks obtained by minimizing the energy in Eq. (1), we compute a pairwise distance matrix $M \in \mathbb{R}^{\mathsf{T}\times\mathsf{T}}$ for each pair as follows. The similarity

$$M_{\mathsf{ij}} = w_{\mathcal{Q}} \cdot \mathcal{Q}_{\mathsf{ij}} + w_{\mathcal{P}} \cdot \mathcal{P}_{\mathsf{ij}} + w_{\mathcal{R}} \cdot \mathcal{R}_{\mathsf{ij}} + w_{\mathcal{S}} \cdot \mathcal{S}_{\mathsf{ij}}, \tag{21}$$

between trajectories i and j is computed based on temporal ($\mathcal{Q}$) and spatial ($\mathcal{P}$) distance, scale ($\mathcal{R}$) and appearance ($\mathcal{S}$) features that are defined as follows: $\mathcal{Q}_{\mathsf{ij}} = \mathsf{s}_{\mathsf{j}} - \mathsf{e}_{\mathsf{i}}$ is the temporal gap between the head (or end point) of track i and the tail (or starting point) of track j. To compute $\mathcal{P}$ and $\mathcal{R}$, both tracks are linearly extrapolated from their termination points and the average of their respective distance in image space is computed at both points in time. Formally,

$$\mathcal{P}_{\mathsf{ij}} = \frac{\|\mathcal{T}_{\mathsf{i}}(\mathsf{s}_{\mathsf{i}}) - \mathcal{T}_{\mathsf{j}}(\mathsf{s}_{\mathsf{i}})\| + \|\mathcal{T}_{\mathsf{i}}(\mathsf{e}_{\mathsf{j}}) - \mathcal{T}_{\mathsf{j}}(\mathsf{e}_{\mathsf{j}})\|}{2} \tag{22}$$

and

$$\mathcal{R}_{\mathsf{ij}} = \frac{|\bar{h}_{\mathsf{i}}(\mathsf{s}_{\mathsf{i}}) - \bar{h}_{\mathsf{j}}(\mathsf{s}_{\mathsf{i}})| + |\bar{h}_{\mathsf{i}}(\mathsf{e}_{\mathsf{j}}) - \bar{h}_{\mathsf{j}}(\mathsf{e}_{\mathsf{j}})|}{2}, \tag{23}$$

where $\bar{h}_{\mathsf{i}}(\cdot)$ is the height of the bounding box of target i. Similarly, the appearance dissimilarity $\mathcal{S}$ is computed as the Bhattacharyya distance between two appearance histograms of the bounding boxes. Tracks are linked in a greedy fashion, until their combined distance exceeds a threshold w_M. Both the weights $w_{\dots}$ for the individual features, as well as the termination threshold w_M are learned via cross validation.

REFERENCES

[1] L. Zhang, Y. Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," in *Proc. IEEE Conf. Comp. Vis. Pattern Recog.*, 2008, pp. 1–8.

[2] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes, "Globally-optimal greedy algorithms for tracking a variable number of objects," in *Proc. IEEE Conf. Comp. Vis. Pattern Recog.*, 2011, pp. 1201–1208.

[3] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua, "Multiple object tracking using K-shortest paths optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1806–1819, Sep. 2011.

[4] A. Milan, S. Roth, and K. Schindler, "Continuous energy minimization for multitarget tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 58–72, Jan. 2014.

[5] A. R. Zamir, A. Dehghan, and M. Shah, "GMCP-Tracker: Global multi-object tracking using generalized minimum clique graphs," in *Proc. Eur. Conf. Comput. Vis.*, vol. 2, 2012, pp. 343–356.

[6] A. Butt and R. Collins, "Multi-target tracking by Lagrangian relaxation to min-cost network flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 1846–1853.

[7] A. Delong, A. Osokin, H. N. Isack, and Y. Boykov, "Fast approximate energy minimization with label costs," *Int. J. Comp. Vis.*, vol. 96, no. 1, pp. 1–27, Jan. 2012.

[8] A. Andriyenko, K. Schindler, and S. Roth, "Discrete-continuous optimization for multi-target tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 1926–1933.

[9] A. Milan, S. Roth, and K. Schindler, "Detection- and trajectory-level exclusion in multiple object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 3682–3689.

[10] D. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Autom. Control*, vol. 24, no. 6, pp. 843–854, Dec. 1979.

[11] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Multi-target tracking using joint probabilistic data association," in *Proc. 19th IEEE Conf. Decision Control Symp. Adaptive Process.*, Dec. 1980, vol. 19, pp. 807–812.

[12] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME–J. Basic Eng.*, vol. 82, no. ser. D, pp. 35–45, 1960.

[13] J. Vermaak, A. Doucet, and P. Pérez, "Maintaining multi-modality through mixture tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2003, pp. 1110–1116.

[14] K. Okuma, A. Taleghani, O. D. Freitas, J. J. Little, and D. G. Lowe, "A boosted particle filter: Multitarget detection and tracking," in *Proc. Eur. Conf. Comput. Vis.*, vol. 1, 2004, pp. 28–39.

[15] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Robust tracking-by-detection using a detector confidence particle filter," in *Proc. 12th Int. Conf. Comput. Vis.*, 2009, pp. 1515–1522.

[16] H. Jiang, S. Fels, and J. J. Little, "A linear programming approach for multiple object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2007, pp. 1–8.

[17] J. Berclaz, F. Fleuret, and P. Fua, "Multiple object tracking using flow linear programming," in *Proc. 12th IEEE Int. Workshop Perform. Eval. Track. Surveillance*, Dec. 2009, pp. 1–8.

[18] P. Lenz, A. Geiger, and R. Urtasun, "FollowMe: Efficient online min-cost flow tracking with bounded memory and computation," *arXiv:1407.6251 [cs]*, Jul. 2014.

[19] J. Henriques, R. Caseiro, and J. Batista, "Globally optimal solution to multi-object tracking with merged measurements," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 2470–2477.

[20] Z. Wu, T. H. Kunz, and M. Betke, "Efficient track linking methods for track graphs using network-flow and set-cover techniques," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, 2011, pp. 1185–1192.

[21] A. Andriyenko and K. Schindler, "Globally optimal multi-target tracking on a hexagonal lattice," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, vol. 1, pp. 466–479.

[22] W. Brendel, M. R. Amer, and S. Todorovic, "Multiobject tracking as maximum weight independent set," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 1273–1280.

[23] A. Dehghan, S. M. Assari, and M. Shah, "GMMCP-tracker: Globally optimal generalized maximum multi clique problem for multiple object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 4091–4099.

[24] S. Oh, S. Russell, and S. Sastry, "Markov chain Monte Carlo data association for general multiple-target tracking problems," in *Proc. 43rd IEEE Conf. Decision Control*, Dec. 2004, vol. 1, pp. 735–742.

[25] B. Benfold and I. Reid, "Stable multi-target tracking in real-time surveillance video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 3457–3464.
[26] H. Ben Shitrit, J. Berclaz, F. Fleuret, and P. Fua, "Tracking multiple people under global appearance constraints," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 137–144.
[27] Z. Khan, T. Balch, and F. Dellaert, "MCMC data association and sparse factorization updating for real time multitarget tracking with merged and multiple measurements," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 1960–1972, Dec. 2006.
[28] M. Andriluka, S. Roth, and B. Schiele, "Monocular 3D pose estimation and tracking by detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2010, pp. 623–630.
[29] B. Yang and R. Nevatia, "An online learned CRF model for multi-target tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 2034–2041.
[30] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "MOTChallenge 2015: Towards a benchmark for multi-target tracking," *arXiv:1504.01942 [cs]*, Apr. 2015.
[31] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, vol. 1, 2005, pp. 886–893.
[32] S. Walk, N. Majer, K. Schindler, and B. Schiele, "New features and insights for pedestrian detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2010, pp. 1030–1037.
[33] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1532–1545, 2014.
[34] B. Leibe, K. Schindler, N. Cornelis, and L. Van Gool, "Coupled detection and tracking from static cameras and moving vehicles," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 10, pp. 1683–1698, Oct. 2008.
[35] J. Ferryman and A. Shahrokni, "PETS2009: Dataset and challenge," in *Proc. 11th IEEE Int. Workshop Perform. Eval. Track. Surveillance*, Dec. 2009, pp. 1–6.
[36] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud, "Two deterministic half-quadratic regularization algorithms for computed imaging," in *Proc. IEEE Int. Conf. Image Process.*, vol. 2, Nov. 1994, pp. 168–172.
[37] L. Ladicky, C. Russell, P. Kohli, and P. H. S. Torr, "Graph cut based inference with co-occurrence statistics," in *Proc. Eur. Conf. Comput. Vis.*, vol. 5, 2010, pp. 239–253.
[38] Y. Boykov, O. Veksler, and R. Zabih. (2001, Nov.). Fast approximate energy minimization via graph cuts, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001.
[39] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer, "Optimizing binary MRFs via extended roof duality," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2007, pp. 1–8.
[40] B. Andres, T. Beier, and J. Kappes, "OpenGM: A C++ library for discrete graphical models," *arXiv:1206.0111*, Jun. 2012.
[41] J. Liu, P. Carr, R. Collins, and Y. Liu, "Tracking sports players with context-conditioned motion models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 1830–1837.
[42] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, Mar. 2012.
[43] V. Kolmogorov, "Convergent tree-reweighted message passing for energy minimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1568–1583, Oct. 2006.
[44] F. R. Kschischang, B. J. Frey, and H.-A. Loelinger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Info. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.
[45] K. Murphy, Y. Weiss, and M. Jordan, "Loopy belief propagation for approximate inference: An empirical study," in *Proc. 15th Conf. Uncertainty Artif. Intell.*, 1999, pp. 467–475.
[46] A. Ess, B. Leibe, K. Schindler, and L. Van Gool, "A mobile vision system for robust multi-person tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–8.
[47] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *Image Video Process.*, vol. 2008, no. 1, pp. 1–10, 2008.
[48] B. Wu and R. Nevatia, "Tracking of multiple, partially occluded humans based on static body part detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, vol. 1, 2006, pp. 951–958.
[49] C.-H. Kuo and R. Nevatia, "How does person identity recognition help multi-person tracking?," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 1217–1224.
[50] A. Milan, L. Leal-Taixé, K. Schindler, and I. Reid, "Joint tracking and segmentation of multiple targets," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 5397–5406.
[51] L. Leal-Taixé, M. Fenzi, A. Kuznetsova, B. Rosenhahn, and S. Savarese, "Learning an image-based motion context for multiple people tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 3542–3549.
[52] J. H. Yoon, M.-H. Yang, J. Lim, and K.-J. Yoon, "Bayesian multi-object tracking using motion context from multiple objects," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2015, pp. 33–40.
[53] C. Dicle, M. Sznaier, and O. Camps, "The way they move: Tracking multiple targets with similar appearance," in *Proc. IEEE Conf. Comput. Vis.*, 2013, pp. 2304–2311.

Anton Milan received the Diplom degree in computer science (Dipl.-Inform.) from the University of Bonn, Germany, in 2008 and the PhD degree (Dr.-Ing.) from the Technische Universität Darmstadt, Germany, in 2013. He has worked as a software developer in the computer graphics industry and currently holds the position as a post-doctoral fellow at the University of Adelaide, Australia. He served as a reviewer for various computer vision conferences and journals. His research interests include visual SLAM and energy-based optimization methods for real-world multiple people tracking scenarios. He is a member of the IEEE.

Konrad Schindler received the Diplomingenieur (Mtech) degree from the Vienna University of Technology, Austria, in 1999, and the PhD degree from the Graz University of Technology, Austria, in 2003. He has worked as a photogrammetric engineer in the private industry, and held researcher positions in the Computer Graphics and Vision Department, Graz University of Technology, the Digital Perception Lab, Monash University, and the Computer Vision Lab, ETH Zurich. He became assistant professor of Image Understanding at the TU Darmstadt, in 2009, and since 2010 has been a tenured professor of photogrammetry and remote sensing at the ETH Zurich. His research interests lie in the field of computer vision, photogrammetry, and remote sensing, with a focus on image understanding and 3D reconstruction. He currently serves as head of the Institute of Geodesy and Photogrammetry, and as an associate editor for the *ISPRS Journal of Photogrammetry and Remote Sensing*, and for the *Image and Vision Computing Journal*. He is a senior member of the IEEE.

Stefan Roth received the Diplom degree in computer science and engineering from the University of Mannheim, Germany, in 2001. In 2003, he received the ScM degree in computer science from the Brown University, and in 2007, the PhD degree in computer science from the same institution. Since 2007, he has been on the faculty of computer science at the Technische Universität Darmstadt, Germany (junior professor 2007-2013, professor since 2013). His research interests include probabilistic and statistical approaches to image modeling, motion estimation, human tracking, and object recognition. He received several awards, including honorable mentions for the Marr Prize at the ICCV 2005 (with M. Black) and ICCV 2013 (with C. Vogel and K. Schindler), the Olympus-Prize 2010 of the German Association for Pattern Recognition (DAGM), and the Heinz Maier-Leibnitz Prize 2012 of the German Research Foundation (DFG). He served as an area chair for the ICCV 2011, ECCV 2012 & 2014, and CVPR 2013, and is member of the editorial board of the *International Journal of Computer Vision (IJCV)*, the *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, and *PeerJ Computer Science*. He is a member of the IEEE.