Discrete-Continuous Optimization for Multi-Target Tracking

Anton Andriyenko[1]   Konrad Schindler[2]   Stefan Roth[1]

[1]Department of Computer Science, TU Darmstadt   [2]Photogrammetry and Remote Sensing Group, ETH Zürich

Abstract

The problem of multi-target tracking is comprised of two distinct, but tightly coupled challenges: (i) the naturally discrete problem of data association, i.e. *assigning image observations to the appropriate target; (ii) the naturally continuous problem of trajectory estimation,* i.e. *recovering the trajectories of all targets. To go beyond simple greedy solutions for data association, recent approaches often perform multi-target tracking using discrete optimization. This has the disadvantage that trajectories need to be pre-computed or represented discretely, thus limiting accuracy. In this paper we instead formulate multi-target tracking as a discrete-continuous optimization problem that handles each aspect in its natural domain and allows leveraging powerful methods for multi-model fitting. Data association is performed using discrete optimization with label costs, yielding near optimality. Trajectory estimation is posed as a continuous fitting problem with a simple closed-form solution, which is used in turn to update the label costs. We demonstrate the accuracy and robustness of our approach with state-of-the-art performance on several standard datasets.*

1. Introduction

Research in multi-target tracking has shown significant progress in recent years. Nevertheless, current algorithms only achieve reasonable performance in comparably easy conditions with only few targets. As soon as the area of interest becomes crowded, the human ability to correctly identify and follow targets – when given sufficient time – still greatly exceeds automatic approaches.

Many of the most successful tracking methods at present perform *tracking by detection*, *i.e.* the target is represented by an object model that can be detected in every frame independently [*e.g.*, 20, 25], in some cases in combination with an online model to deal with lighting and appearance variation [*e.g.*, 11]. The advantages of using an object detector are that it naturally handles re-initialization if a target has been lost, and that it avoids excessive model drift [7]. The detector yields the per-frame evidence for the presence of a target. Hence, when dealing with a single target, tracking amounts to fitting a single temporally consistent trajectory such that it optimally accounts for that evidence. In the multi-target case the task is significantly more difficult, since the issue of *data association* must be addressed at the same time. Intuitively speaking, one has to establish a unique identity for each target, and then simultaneously estimate the motion patterns of all targets and the assignment of detections to the targets.

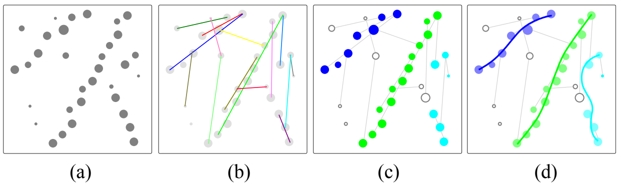

Figure 1. Given a number of unlabeled object detections *(a)* and a number of possible trajectory hypotheses *(b)*, our method labels all detections *(c)* and re-estimates the trajectories *(d)* using an alternating discrete-continuous optimization scheme.

This poses a number of difficult challenges. To start with, the number of targets is usually unknown and may vary over time. In addition, the detector output is only partly reliable, thus one has to account for missing evidence (false negatives), as well as incorrect evidence (false alarms). The task is further complicated by the fact that unless targets always remain well separated, the space of possible trajectories grows exponentially over time. Furthermore, trajectories should obey certain constraints, such as that two targets cannot be at the same location at the same time. Addressing these challenges requires coping with two distinct, but tightly coupled modeling issues. Labeling each detection as either belonging to a certain target or being a false alarm is intrinsically in the *discrete domain*. For reasonable interpretations of the observed scene, the same detection can only have a single label. However, the target locations over time are naturally described in a *continuous state space* (this may also include further dimensions such as size, velocity, *etc.*).

Existing techniques strike the balance between the two tasks in different ways. An extensive body of recent work focuses on data association and uses powerful discrete optimization algorithms to approach this NP-hard problem. However, the continuous aspect of trajectory estimation suf-

To appear in the Proc. of the *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Providence, RI, June 2012.

fers, either because trajectories have to be pre-computed in absence of any data association [26, 27], or the trajectories are spatially discretized [2, 4]. Other techniques focus on trajectory estimation in a continuous state space, but limit the data association to a choice from a pre-computed set of potential labelings [17]. Sampling-based approaches [14, 19] have attempted to build a bridge between the discrete and continuous aspects, but remain relatively limited in the expressiveness of the underlying model.

In this paper we formulate data association and trajectory estimation jointly as the *minimization of a consistent discrete-continuous energy*, which treats each aspect in its natural domain. To that end we build on recent advances in multi-model fitting introduced by Delong *et al.* [9]. We show how to formulate multi-target tracking in that framework and extend the inference algorithm accordingly. Trajectories are modeled by piecewise polynomials, which can be fitted to a set of target hypotheses in closed form. Given these trajectories, data association is updated by α-expansion, taking into account global trajectory properties such as the dynamics and persistence of moving objects through individual *label costs*. The two steps are alternated to minimize a single discrete-continuous objective, such that trajectory estimation can take advantage of data association and vice versa (*cf*. Fig. 1).

The present work thus makes the following contributions: *(i)* we formulate multi-target tracking as the minimization of a unified discrete-continuous energy; *(ii)* we demonstrate the applicability of the label-cost framework to the tracking problem; and *(iii)* extend this scheme to take into account the problem specifics, where measuring the goodness of a trajectory goes beyond the geometric fitting residual. To the best of our knowledge, this paper is the first to pose tracking as discrete-continuous optimization with label costs. As our experiments on various standard datasets indicate, this substantially increases the tracking accuracy while retaining the benefits of performing non-greedy data association.

2. Related Work

Tracking has been an active research topic in computer vision and other fields for several decades. In this review we thus concentrate on recent advances in visual multi-target tracking.

Multi-object tracking methods can be divided into two categories. The first only relies on the information from past frames to estimate the current state *recursively*. While early Kalman filtering approaches [21] only model linear target motion, more recent sample-based filters, such as particle filtering [6, 14], can deal with more complex multi-modal posteriors. However, the number of particles needed to accurately approximate the posterior in complex situations grows quickly and is hard to handle in practice.

The second category allows for a certain latency and *globally* solves for all trajectories within a given time window. In this case, it is common practice to restrict the optimization to a finite state space. One way to do this is to restrict the set of possible object locations, such as by requiring trajectories to exactly pass either through the individual detections [13], or through a set of pre-computed tracklets [26, 27]. The (near) optimal solution can then be found by linking the detections and tracklets by max-flow computation. A slightly different approach is presented in [17, 18], where a redundant set of putative trajectories is pre-computed, and the optimization takes place at the trajectory level by pruning to an optimal subset, formalized as a quadratic Boolean problem.

A different way to reduce the complexity is to subdivide the tracking area into disjoint, locally fully connected cells. Object motion is then described by binary occupancy variables for those cells, and the resulting problems are solved to (near) global optimality using LP-relaxation [2, 4].

Somewhat against the trend, [3] also belongs to the second group of non-recursive trackers, but relaxes all discrete variables to a completely continuous state space. This, however, results in a highly non-convex optimization with many local minima, which necessitate a heuristic energy minimization scheme with repeated jump moves.

Here, we aim for a mixed discrete-continuous formulation, which we feel is a more natural way to describe the situation: data association between target detections and trajectories is kept discrete, nonetheless trajectory fitting is performed in the continuous domain without artificially restricting the state space. The proposed formulation allows to improve target locations compared to the – necessarily noisy – detection evidence, and yields smooth target dynamics. Nevertheless, the data association continues to be amenable to well-established discrete optimization techniques for labeling problems, such as graph cuts [5, 16] and (tree-reweighted) belief propagation [*e.g.*, 15]. In contrast to previous discrete-continuous approaches based on Markov Chain Monte Carlo (MCMC) sampling [14, 19], the label cost framework makes it rather easy to incorporate global trajectory properties into the formulation, such as high-order data fidelity, which penalizes trajectories that do not pass near detections for extended periods.

3. Discrete-Continuous Multi-Object Tracking

In agreement with the majority of recent multi-target tracking methods [*e.g.*, 1, 4, 18, 27], we pursue tracking by detection. Targets (here, pedestrians) are separated from the background in a preprocessing step and form a set of target hypotheses, which are then used to infer the targets' trajectories. We thus run a sliding window detector, based on SVM classification of histograms of oriented gradients (HOG) [8] and relative optical flow (HOF) [24]. The de-

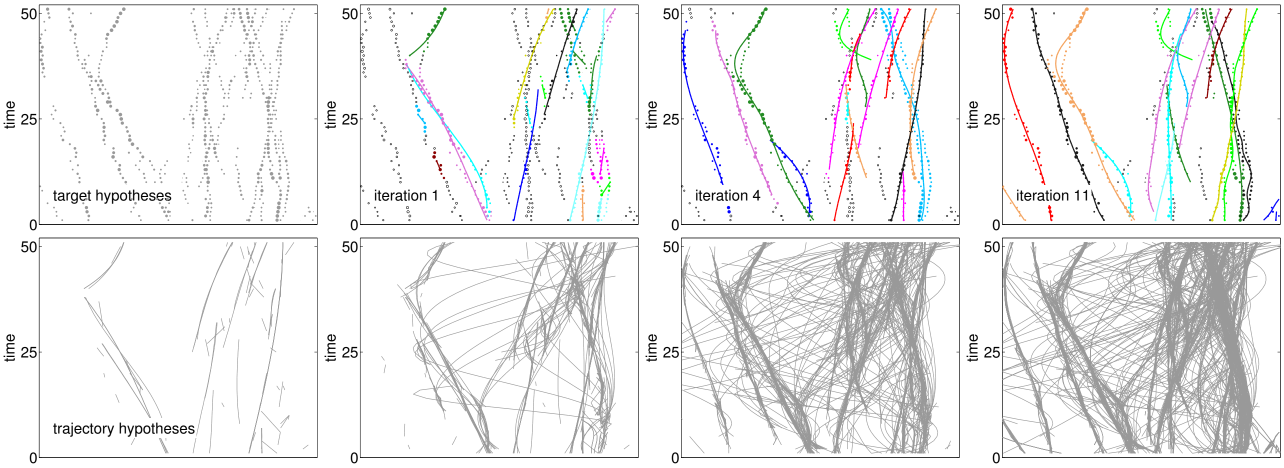

Figure 2. Starting from a set of object detections and trajectory hypotheses (left column), our algorithm performs data association and trajectory estimation by alternating between solving a multi-labeling problem, and minimizing a convex, continuous energy. The current set of trajectory hypotheses at each iteration is shown in the second row.

tector yields a set of target hypotheses $\mathbf{D}$. We denote the $\mathtt{j}^{\text{th}}$ detection at time $\mathtt{t} \in \{1, \ldots, \mathtt{T}\}$ as $\mathtt{d}_\mathtt{j}^\mathtt{t}$, its location as $p_\mathtt{j}^\mathtt{t} \in \mathbb{R}^2$ and $c_\mathtt{j}^\mathtt{t}$ its confidence. If the camera calibration is available and a reliable depth estimate can be obtained, the $p_\mathtt{j}^\mathtt{t}$ represent (x, y)-coordinates on a ground plane. Otherwise, they correspond to pixel coordinates on the image plane. To emphasize the distinction between discrete and continuous variables, we write discrete ones in typewriter font $(\mathtt{a}, \mathtt{b}, \ldots)$ and continuous ones in italics $(a, b, \ldots)$. Discrete sets are denoted with bold capitals $(\mathbf{A}, \mathbf{B}, \ldots)$ and continuous ones with calligraphic letters $(\mathcal{A}, \mathcal{B}, \ldots)$.

Given the set of target hypotheses $\mathbf{D}$, our goal is to identify a set of target trajectories $\mathcal{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_\mathtt{N}\}$. This implies that we also need to search for a data association $\mathtt{f}$, which for each detection $\mathtt{d} \in \mathbf{D}$ assigns a label $\mathtt{f}_\mathtt{d} \in \mathbf{L} = \{1, \ldots, \mathtt{N}\} \cup \varnothing$. Thereby a detection is either identified as belonging to one of the trajectories or, using the additional outlier label $\varnothing$, identified as a false alarm. We eventually aim to perform multi-target tracking by minimizing a joint energy $E(\mathcal{T}, \mathtt{f})$ w.r.t. the trajectories $\mathcal{T}$ and the data association $\mathtt{f}$. To ease understanding, we first introduce the component energies and unify them later.

3.1. Continuous trajectory model

In contrast to purely discrete approaches to multi-target tracking [2, 4], we represent individual trajectories in continuous space and use cubic B-splines for that purpose. 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 intrinsic motion smoothness. More specifically, the spline for each trajectory $\mathcal{T}_\mathtt{i} : t \in \mathbb{R}_0^+ \to (x, y)^\mathrm{T} \in \mathbb{R}^2$ describes the target location $(x, y)^\mathrm{T}$ for each point in time t. We assume that the spline has a varying number $\mathtt{c}_\mathtt{i}$ of control points and is parametrized by a coefficient matrix $C_\mathtt{i} \in \mathbb{R}^{2\mathtt{c}_\mathtt{i} \times 4}$. We found that it is advantageous to explicitly model the temporal starting points $\mathtt{s}_\mathtt{i}$ and end points $\mathtt{e}_\mathtt{i}$ of each trajectory ($t \in [\mathtt{s}_\mathtt{i} - \Delta, \mathtt{e}_\mathtt{i} + \Delta]$), because the splines tend to take on extreme values outside their support otherwise, which results in highly unlikely motion patterns. To ensure that the spline does not take on extreme values immediately outside of $[\mathtt{s}, \mathtt{e}]$, which would prevent other detections in adjacent frames from being assigned to the trajectory later, we add a safety margin of Δ on either side.

If we for now suppose that we are already given a data association $\mathtt{f}$, we can formulate the trajectory estimation problem as minimization of the energy

$$E_\mathtt{f}^{\text{te}}(\mathcal{T}) = \sum_{\mathtt{i}=1}^{\mathtt{N}} \left(E_\mathtt{f}^{\text{te}}(\mathcal{T}_\mathtt{i}) + \hat{E}_v^{\text{te}}(\mathcal{T}_\mathtt{i}) \right), \tag{1}$$

where $E_\mathtt{f}^{\text{te}}(\mathcal{T}_\mathtt{i})$ models how well trajectory $\mathcal{T}_\mathtt{i}$ fits to the hypotheses assigned by $\mathtt{f}$ and $\hat{E}_v^{\text{te}}(\mathcal{T}_\mathtt{i})$ models the smoothness of $\mathcal{T}_\mathtt{i}$ on the safety margin. For each trajectory we aim to minimize the weighted Euclidean distance to each assigned target hypothesis in all valid frames:

$$E_\mathtt{f}^{\text{te}}(\mathcal{T}_\mathtt{i}) = \sum_{\mathtt{t}=\mathtt{s}_\mathtt{i}}^{\mathtt{e}_\mathtt{i}} \sum_{\mathtt{j}=1}^{|\mathbf{D}^\mathtt{t}|} \delta\left[\mathtt{i} - \mathtt{f}_{\mathtt{d}_\mathtt{j}^\mathtt{t}}\right] \cdot c_\mathtt{j}^\mathtt{t} \cdot \left\| p_\mathtt{j}^\mathtt{t} - \mathcal{T}_\mathtt{i}(\mathtt{t}) \right\|^2, \tag{2}$$

where $|\mathbf{D}^\mathtt{t}|$ is the number of detections in frame $\mathtt{t}$. The Kronecker delta ($\delta[\mathtt{a} - \mathtt{b}] = 1$ if $\mathtt{a} = \mathtt{b}$, and 0 otherwise) ensures that only target hypotheses $\mathtt{d}_\mathtt{j}^\mathtt{t}$ are counted that are assigned to trajectory $\mathtt{i}$. On the safety margin the spline is fit to virtual locations $v_\mathtt{i}^\mathtt{t}$ obtained by linear extrapolation:

$$\hat{E}_v^{\text{te}}(\mathcal{T}_\mathtt{i}) = \sum_{\substack{\mathtt{s}_\mathtt{i} - \Delta \leq \mathtt{t} < \mathtt{s}_\mathtt{i} \\ \mathtt{e}_\mathtt{i} < \mathtt{t} \leq \mathtt{e}_\mathtt{i} + \Delta}} \left\| v_\mathtt{i}^\mathtt{t} - \mathcal{T}_\mathtt{i}(\mathtt{t}) \right\|^2. \tag{3}$$

In all our experiments we use $\Delta = 2$. A convenient property of this cubic B-spline formulation is that minimizing Eq. (1) amounts to solving a weighted least squares problem, which can be done in a globally optimal fashion in closed form.

3.2. Discrete data association

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

A large class of labeling problems in computer vision are formulated in terms of the minimization of an energy of a discrete, pairwise Markov random field. This also serves as the starting point here. To that end we identify each individual detection $\mathtt{d} \in \mathbf{D}$ with a vertex of the graph $\mathbf{G} = (\mathbf{D}, \mathbf{E})$. Furthermore, all pairs of detections in adjacent frames whose distance is below a threshold τ are connected by an edge (*cf*. Fig. 3):

$$\mathbf{E} = \left\{ \left(\mathtt{d}_\mathtt{j}^\mathtt{t}, \mathtt{d}_\mathtt{k}^{\mathtt{t}+1}\right) \,\middle|\, \left\|p_\mathtt{j}^\mathtt{t} - p_\mathtt{k}^{\mathtt{t}+1}\right\| < \tau, \mathtt{t} = 1, \dots, \mathtt{T}-1 \right\}.$$

The motivation for this is that nearby detections in adjacent frames should be encouraged to have the same trajectory label. We refrain from longer-range 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. Overall, this gives rise to the discrete pairwise MRF energy

$$E_\mathcal{T}^\mathrm{da}(\mathtt{f}) = \sum_{\mathtt{d} \in \mathbf{D}} U_\mathtt{d}(\mathtt{f}_\mathtt{d}, \mathcal{T}) + \sum_{(\mathtt{d},\mathtt{d}') \in \mathbf{E}} S_{\mathtt{d},\mathtt{d}'}(\mathtt{f}_\mathtt{d}, \mathtt{f}_{\mathtt{d}'}), \quad (4)$$

consisting of a unary or data term $U_\mathtt{d}$ for each vertex (detection) and a pairwise smoothness term $S_{\mathtt{d},\mathtt{d}'}$ for each edge. Chains of vertices linked by such pairwise potentials can be viewed as probabilistic "soft tracklets".

While minimizing the energy in Eq. (4) w.r.t. the labeling $\mathtt{f}$ is in general NP-hard, globally optimal solutions can be found in polynomial time for binary, submodular problems [16]. Moreover, well-proven approximate inference algorithms exist for the multi-label case [*e.g*., 5] and the non-submodular case [*e.g*., 15, 22].

Data term. As usual, the data term is responsible for keeping the solution close to the observed data. To stay consistent with Eq. (2), we use the squared Euclidean distance between the detection location $p_\mathtt{j}^\mathtt{t}$ and its associated trajectory $\mathcal{T}_\mathtt{l}$, weighted by the detection confidence $c_\mathtt{j}^\mathtt{t}$:

$$U_{\mathtt{d}_\mathtt{j}^\mathtt{t}}(\mathtt{l}, \mathcal{T}) = c_\mathtt{j}^\mathtt{t} \cdot \|p_\mathtt{j}^\mathtt{t} - \mathcal{T}_\mathtt{l}(\mathtt{t})\|^2. \quad (5)$$

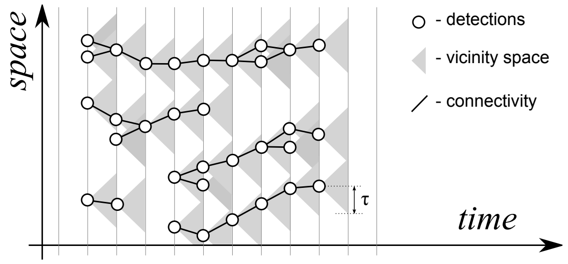

Figure 3. Neighborhood structure of the underlying pairwise Markov random field. Detections in adjacent frames are connected if their distance is below a certain threshold.

If the detection is labeled as an outlier, it is penalized with a constant outlier cost O, again modulated by $c_\mathtt{j}^\mathtt{t}$:

$$U_{\mathtt{d}_\mathtt{j}^\mathtt{t}}(\varnothing, \mathcal{T}) = c_\mathtt{j}^\mathtt{t} \cdot O. \quad (6)$$

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. The data term incorporates this by penalizing a larger distance to a weak detection less than to a confident one (Eq. (5)). The weight of the outliers is similarly reduced (Eq. (6)), so as to promote false detections being labeled as outliers.

Smoothness term. The pairwise terms connect spatio-temporal neighbors and favor consistent labelings between them based on a simple generalized Potts potential:

$$S_{\mathtt{d}_\mathtt{j}^\mathtt{t}, \mathtt{d}_\mathtt{k}^{\mathtt{t}+1}}(\mathtt{f}_{\mathtt{d}_\mathtt{j}^\mathtt{t}}, \mathtt{f}_{\mathtt{d}_\mathtt{k}^{\mathtt{t}+1}}) = \eta \cdot \delta\left[\mathtt{f}_{\mathtt{d}_\mathtt{j}^\mathtt{t}} - \mathtt{f}_{\mathtt{d}_\mathtt{k}^{\mathtt{t}+1}}\right]. \quad (7)$$

3.3. Discrete-continuous tracking with label costs

Due to the choice of formulations for both trajectory estimation and data association, it is now possible to unify them in a single, consistent energy function:

$$\begin{aligned} E(\mathcal{T}, \mathtt{f}) = &\sum_{\mathtt{d} \in \mathbf{D}} U_\mathtt{d}(\mathtt{f}_\mathtt{d}, \mathcal{T}) + \sum_{(\mathtt{d},\mathtt{d}') \in \mathbf{E}} S_{\mathtt{d},\mathtt{d}'}(\mathtt{f}_\mathtt{d}, \mathtt{f}_{\mathtt{d}'}) \\ &+ \sum_{\mathtt{i}=1}^{\mathtt{N}} \hat{E}_v^\mathrm{te}(\mathcal{T}_\mathtt{i}) + \kappa \cdot h_\mathtt{f}(\mathcal{T}). \end{aligned} \quad (8)$$

To understand this formulation, it is instructive to first consider the case when the last term is not active (*i.e*. $\kappa = 0$). In this case minimizing Eq. (8) w.r.t. the trajectories $\mathcal{T}$ given a fixed labeling $\mathtt{f}$ is equivalent to trajectory estimation, *i.e*. minimizing Eq. (1), and minimizing it w.r.t. the labeling $\mathtt{f}$ given fixed trajectories $\mathcal{T}$ is equivalent to data association, *i.e*. minimizing Eq. (4). However, alternating minimization of such an objective will not lead to the desired result. The most obvious problem (but not the only one) is that neither of the two parts includes a model selection term to regularize the number of trajectories. Given the variable number of targets, the alternation would thus overfit by instantiating more trajectories to reduce the fitting error.

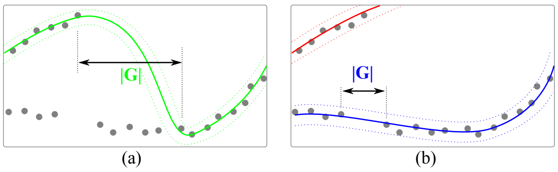

Figure 4. The high-order data fidelity h^{fid} addresses the problem of long time spans during which a trajectory has no nearby detections *(a)*. The blue trajectory in *(b)* has a much lower label cost.

To overcome the problem we follow the recent work of Delong *et al.* [9] and rely on a so-called label cost term $h_{\mathtt{f}}(\mathcal{T})$, which specifies a cost that is applied to each label and takes effect as long as the labeling contains this label at least once. More specifically, our label cost term $h_{\mathtt{f}}(\mathcal{T})$

- integrates a dynamic model and keeps trajectories within physical limits,
- enforces long, persistent trajectories, by penalizing long sections of missing evidence, as well as tracks that start or end far from the image border,
- discards mutually competing hypotheses that cannot exist simultaneously, and finally
- penalizes the total number of current targets.

We now turn to the individual components of the label cost.

Dynamics. In real world tracking applications, some prior information is usually available about the targets' motion. Most importantly, their velocity is bounded by physical constraints. We therefore impose a penalty on the cubic coefficient of the splines, which carries the predominant influence on the maximal velocity. The resulting label cost for trajectory $\mathcal{T}_{\mathtt{i}}$ is defined as

$$h_{\mathtt{i}}^{\text{dyn}} = \lambda \cdot \max_{\mathtt{r}} C_{\mathtt{i}}(\mathtt{r}, 1). \qquad (9)$$

Persistence. Enforcing long, persistent trajectories is key to avoiding unnecessary identity switches. Our spline representation allows us to identify the start and end points of each trajectory and impose a higher penalty on those that initiate or terminate far from the image border:

$$h_{\mathtt{i}}^{\text{per}} = \mu \cdot \Big(\bar{\mathtt{b}}\big(\mathcal{T}_{\mathtt{i}}(\mathtt{s}_{\mathtt{i}})\big) + \bar{\mathtt{b}}\big(\mathcal{T}_{\mathtt{i}}(\mathtt{e}_{\mathtt{i}})\big) \Big) + \nu \cdot (\mathtt{e}_{\mathtt{i}} - \mathtt{s}_{\mathtt{i}})^{-1}, \quad (10)$$

where $\bar{\mathtt{b}}(\cdot)$ denotes the distance to the image border. The last term penalizes short trajectories.

High-order data fidelity. The unary data term from Eq. (5) encourages trajectories to pass near the detections to better explain the observed image evidence. In practice we find that this alone frequently leads to trajectories that do not pass near any detection for extended periods (*cf*. Fig. 4(a)). While the model should allow for such gaps to be able to handle temporary target occlusions (Fig. 4(b)), it is important to penalize gaps that are too large. This aspect cannot be trivially incorporated into the unary data term as it requires the entire trajectory to be considered; we therefore integrate it into the label cost. Trajectories that are far away from detections over longer time spans are assigned a higher cost than those that are continually near detections:

$$h_{\mathtt{i}}^{\text{fid}} = \xi \cdot \sum_{\mathtt{k}} |\mathbf{G}_{\mathtt{k}}|^3, \qquad (11)$$

where the $\mathbf{G}_{\mathtt{k}}$ are the sets of all consecutive frames in which the trajectory $\mathcal{T}_{\mathtt{i}}$ does not pass near any detection (no matter whether these detections are assigned to $\mathcal{T}_{\mathtt{i}}$ or not, otherwise one would already have to know the data association).

Mutual exclusion. A further aspect of multi-target tracking is collision avoidance. The most natural approach may seem to formulate collision avoidance as a pairwise term – if two putative target locations are close to each other, add a high penalty unless at least one of them is labeled as an outlier. Unfortunately this complicates inference considerably, because repulsive edge potentials that favor two nodes having different labels are supermodular. While there are approximate inference algorithms that can deal with supermodular terms [15, 22], they still tend to lead to better approximate solutions for submodular energies.

Instead, we incorporate collision handling into the label cost. To this end, the minimal distance between all pairs of trajectories at the time where they are closest to each other is computed and used to define the label cost:

$$h_{\mathtt{i}}^{\text{col}} = \zeta \cdot \Big(\min_{\mathtt{j}<\mathtt{i}} \min_{\mathtt{t}\in\mathbf{O}} \|\mathcal{T}_{\mathtt{i}}(\mathtt{t}) - \mathcal{T}_{\mathtt{j}}(\mathtt{t})\| \Big)^{-1}, \qquad (12)$$

where $\mathbf{O} = \{\max(\mathtt{s}_{\mathtt{i}}, \mathtt{s}_{\mathtt{j}}), \ldots, \min(\mathtt{e}_{\mathtt{i}}, \mathtt{e}_{\mathtt{j}})\}$ is the temporal overlap of the two trajectories. An intuitive interpretation is that a configuration where two trajectories $\mathcal{T}_{\mathtt{i}}$ and $\mathcal{T}_{\mathtt{j}}$ are too close is highly unlikely or even physically impossible. In this case, the discrete optimization procedure will choose to abandon the one with the higher ID (*i.e.* $\mathcal{T}_{\mathtt{i}}$) because of its high label cost. This way trajectory hypotheses that were initially proposed earlier are favored over more recently generated ones (*cf*. Sec. 3.4).

Regularization. Finally, a constant regularization cost $h_{\mathtt{i}}^{\text{reg}} = 1$ is used to penalize too many existing trajectories.

Full label cost. The entire label cost is thus defined as

$$h_{\mathtt{f}}(\mathcal{T}) = \sum_{\substack{\mathtt{i}=1 \\ \exists \mathtt{d}:\mathtt{f}_{\mathtt{d}}=\mathtt{i}}}^{\mathtt{N}} h_{\mathtt{i}}^{\text{dyn}} + h_{\mathtt{i}}^{\text{per}} + h_{\mathtt{i}}^{\text{fid}} + h_{\mathtt{i}}^{\text{col}} + h_{\mathtt{i}}^{\text{reg}}. \qquad (13)$$

Note that the weighting relative to the unary and pairwise terms is controlled by κ, see Eq. (8). To understand the effect of Eq. (13) it is important to realize that the cost is only incurred for those trajectories that have at least one detection assigned to them.

3.4. Optimization

While optimization with label costs is challenging due to the fact that they are global terms, it can be approached using the integrated energy minimization framework of [9, 12]. To that end we alternate between minimizing Eq. (8) w.r.t. $\mathtt{f}$ and $\mathcal{T}$. Data association, *i.e.* minimization w.r.t. $\mathtt{f}$, thereby benefits from a seamless integration of the label costs into the well studied α-expansion framework with graph cuts, because the energy function remains submodular. This not only leads to strong local optima in practice, but also guarantees a bounded optimality gap (see [9] for details regarding the theoretical properties). Trajectory estimation, *i.e.* minimization w.r.t. $\mathcal{T}$, is somewhat more challenging because the label cost is difficult to optimize w.r.t. the trajectories $\mathcal{T}_{\mathtt{i}}$. To cope with this, we temporarily disregard the label cost, perform least squares minimization of the remaining terms for each individual $\mathcal{T}_{\mathtt{i}}$ and verify that this actually reduces the overall energy, including the label cost. If the overall energy with label cost is not reduced, the previous trajectory is retained. The energy from Eq. (8) can thus only decrease or stay the same.

The motivation is the following: on one hand, the simplified minimization is convex and can be carried out efficiently in closed form, yet is guaranteed to never increase the energy. On the other hand, the simplification should have only a small effect in the context of the complete optimization scheme: near good minima of the energy the gradient of $h_{\mathtt{f}}(\mathcal{T})$ will be small, because the solution already obeys the physical constraints of Sec. 3.3; far from the minima, a large $\frac{\partial}{\partial \mathcal{T}} h_{\mathtt{f}}(\mathcal{T})$ would mean that a different path of the trajectories would be physically a lot more plausible while still staying close to the evidence, in which case it is likely to be picked up by the hypothesis expansion (see below). We thus prefer to defer the difficult aspects of the energy to subsequent iterations of the discrete optimization.

Generating initial trajectory hypotheses. The optimization is bootstrapped with an initial set of trajectory hypotheses obtained in two ways: We use RANSAC to fit trajectories 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, as well as trajectories that pass near more detections. Additionally, we generate candidate trajectories using an extended Kalman filter (EKF) initialized at all detections and using a variety of parameters. 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.[1]

Expanding 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 iteration, based on the current solution. Note that additional hypotheses do not change the nature of the energy; solutions in the expanded space can only have equal or lower energy.

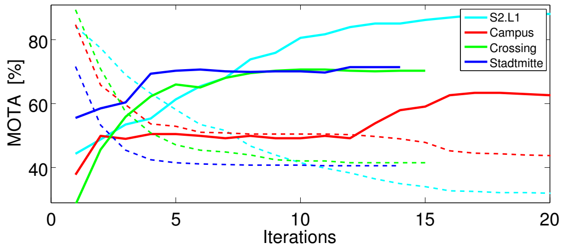

Figure 5. Convergence of the optimization. The energy keeps decreasing for 5-20 iterations (dashed lines, rescaled for visualization), and this is reflected in the tracking accuracy (solid lines).

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; *(2)* existing trajectories are expanded in time or split in regions with no detections; *(3)* pairs of existing trajectories are merged into new ones as long as their combination results in a physically plausible motion; *(4)* splines with a higher number of control points are added on top of currently active ones. Note that in all cases existing 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 energy are removed from the hypothesis space, which guarantees that active hypotheses are never removed.

Implementation details. Although all components of the label cost can be weighted individually, we found that in the majority of settings the results remain stable. Empirically, the penalty ζ for overlapping trajectories can be set to 0. The overlap is rather expensive to compute, whereas in our experience the regularization term h^{reg} already penalizes duplicate trajectories such that explicitly modeling exclusion does not improve performance. To reduce the effect of random sampling, we run the optimization with five different random seeds and pick the result with the lowest energy. The convergence behavior is shown in Fig. 5. Note that although the most significant performance boost usually appears within the first few iterations, the optimization scheme is still able to find better results in later expansion steps. Our current MATLAB code takes ~ 0.5s per frame to converge (excluding the object detector). With an optimized implementation real-time performance is within reach.

4. Experiments

We evaluate our method on four publicly available video sequences. Three sequences (*Campus*, *Crossing* and *Stadtmitte*) are taken from the TUD dataset [1]. The videos are 91, 201 and 179 frames long and show walking pedestrians in a city environment. Due to the low viewpoint, targets fre-

[1] It is a common observation that α-expansion is largely independent of the initialization, unless the unaries are very weak.

Table 1. Average performance over four datasets (see text).

2D perf.	MOTA	MOTP	FPR	FNR	ID Sw.
detector	–	–	38.6 %	27.2 %	–
baseline1	39.8 %	76.0 %	2.4 %	57.6 %	12.8
baseline2	61.1 %	74.7 %	9.7 %	29.1 %	10.2
baseline3	62.0 %	76.7 %	9.7 %	28.2 %	7.0
our method	71.4 %	74.7 %	4.4 %	24.1 %	7.0

Table 2. Comparison of our proposed method to two state-of-the-art trackers on PETS'09 S2.L1. The results of [4, 6] were extracted from Fig. 3 in [10] and are therefore rounded to the closest integer.

2D performance	MOTA	MOTP	MODA	MODP
Berclaz *et al.* [4]	82 %	56 %	85 %	57 %
Breitenstein *et al.* [6]	75 %	**60 %**	89 %	**60 %**
our method	**89.3%**	56.4%	**90.8%**	57.3%

quently become occluded for several frames and their size in the image varies significantly. Note that although we do not explicitly handle occlusions, our method is able to connect the correct trajectories across occlusion gaps in most cases. The low viewpoint makes it hard to correctly estimate target locations on the ground plane. We thus prefer to perform tracking in image space for these sequences. Additionally, we evaluate on the first view of the S2.L1 sequence from the PETS 2009/2010 benchmark. This video of 795 frames, recorded from a distant viewpoint, has become a de facto standard for benchmarking multi-target tracking.

For the quantitative evaluation we rely on the widely used CLEAR MOT metrics [23]. The *Multi-Object Tracking Accuracy* (MOTA) combines all errors (missed targets, false alarms, identity switches) into one number, normalized to the range 0..100 %. A match between the tracker output and the ground truth is defined as $>$ 50% intersection-over-union of their bounding boxes. The related *Multi-Object Detection Accuracy* (MODA) only checks for missed targets and false alarms, but does not penalize trajectories switching from one target to another. The *Multi-Object Tracking Precision* (MOTP) averages the bounding box overlap over all tracked targets as a measure of localization accuracy, whereas the closely related MODP averages the overlap over all frames. Moreover, we also report the false positive (FPR) and false negative rates (FNR), as well as the number of identity switches (ID Sw.). Finally, for a direct comparison with [3] we report the number of mostly tracked (MT) and mostly lost (ML) trajectories, track fragmentations (FM), and ID switches.

Table 3. Comparison of our approach to the purely continuous framework of [3] using their publicly available ground truth.

3D performance	MOTA	MOTP	MT	ML	FM	ID Sw.
TUD-Stadtmitte	**61.8%**	63.2%	**6**	**0**	**1**	**4**
[3]	60.5%	**65.8%**	**6**	**0**	4	7
PETS'09 S2.L1	**95.9%**	**78.7%**	**22**	**0**	**8**	**10**
[3]	81.4%	76.1%	19	**0**	21	15

We compare our method to various baselines (Tab. 1), where we replace the α-expansion step by a greedy labeling algorithm based on hypotheses from RANSAC. Given an initial set of trajectory hypotheses (the same as in Sec. 3.4), the *baseline1* algorithm chooses the one with the lowest cost (based on a truncated Euclidean distance to all detections and separately tuning the threshold for better performance). All detections within the threshold are then removed and the next best trajectory is identified (similar to [1]). One issue of this greedy strategy is that the number of targets would grow until all detections have been explained by at least one trajectory. To prevent this, we only allow a trajectory to become active if the number of detections within the threshold is large enough. As expected, greedy data association quickly gets stuck in a local minimum and is not able to recover from this, which results in a large number of short trajectories. To improve this baseline, we enlarge the initial set with all trajectory hypotheses extracted from the final iteration of our discrete-continuous optimization (*baseline2*), and finally even with the ground truth trajectories (*baseline3*). We are still able to outperform this "cheating" baseline by 9.4 percentage points even when the correct trajectories are available for greedy model selection.

Next, we compare to several state-of-the-art trackers. To assess the benefits of the proposed formulation, we compare to the entirely discrete formulation of Berclaz *et al.* [4], the entirely continuous formulation of Andriyenko and Schindler [3], as well as the recent particle filtering method of Breitenstein *et al.* [6]. Of these methods, [3] has been evaluated on the ground plane in 3D space, whereas the two others have published results in 2D image space.

Table 2 shows a comparison to the available 2D results. The outputs of all trackers, including ours, were evaluated by the PETS organizers using their testing protocol and withheld ground truth. In terms of MOTA we outperform the (nearly) globally optimal discrete method of Berclaz *et al.* [4] by $\sim$ 7 percentage points, and the particle filtering framework of Breitenstein even by $\sim$ 14 percentage points, although compared to the latter the precision is slightly lower. Note that the gap in detection accuracy (MODA, not counting identity switches) is smaller, an indication that the improvement is indeed due to better data association.

To compare with the continuous formulation of [3] we use their publicly available ground truth data (see Tab. 3). Note that the 2D and 3D "ground thruths" were annotated independently, and that the 3D evaluation requires a target radius in 3D world units (defined in [3] to be 1 m). The results thus differ. In the 3D evaluation, our method again achieves clearly better performance, tracking more targets and significantly reducing the number of track fragmentations and ID switches. The tracking precision on the TUD-Stadtmitte dataset is slightly lower. We note though that the low camera viewpoint makes precise 3D estimation rather

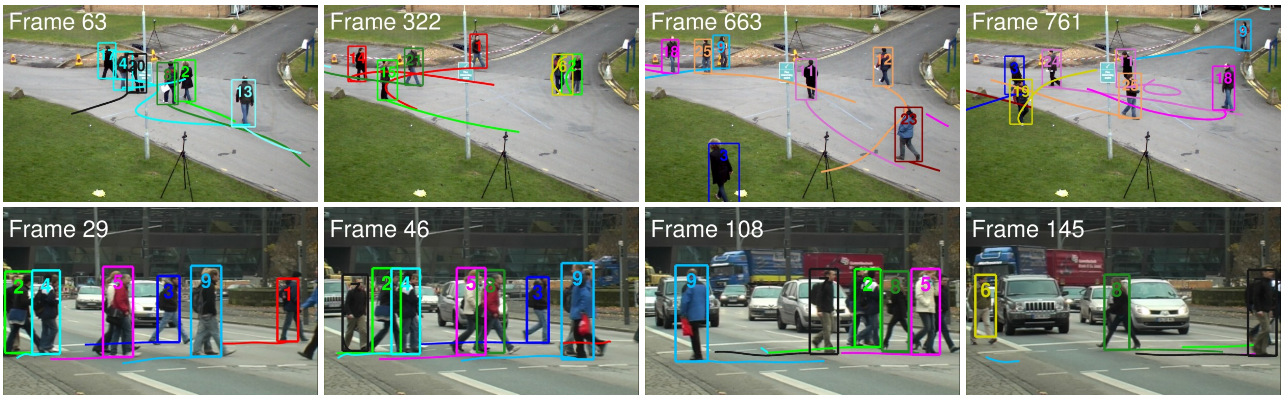

Figure 6. Example frames from our discrete-continous energy minimization approach on the *PETS'09 S2.L1* and *TUD-Crossing* datasets. People are successfully tracked over long time periods (depicted by corresponding trails) while preserving their identities.

difficult (for both the tracker and the annotator). Fig. 6 illustrates our tracking results. Note that our method shows robust performance independent of viewpoint and target size.

5. Conclusion and Future Work

We presented a global multi-target tracking approach that jointly addresses data association and trajectory estimation by minimizing a consistent discrete-continuous energy. The method proceeds iteratively by solving data association to (near) global optimality by α-expansion with label costs, and analytically fitting continuous trajectories to the assigned detections. We demonstrated that the proposed formulation outperforms greedy data association, as well as both discrete and continuous state-of-the-art trackers. In future work we plan to explore alternative labeling algorithms to go beyond submodular pairwise terms.

References

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

[2] A. Andriyenko and K. Schindler. Globally optimal multi-target tracking on a hexagonal lattice. *ECCV*, 2010.

[3] A. Andriyenko and K. Schindler. Multi-target tracking by continuous energy minimization. *CVPR*, 2011.

[4] J. Berclaz, F. Fleuret, and P. Fua. Multiple object tracking using flow linear programming. *Winter-PETS*, 2009.

[5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11), 2001.

[6] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. *PAMI*, 33(9), 2011.

[7] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. *CVPR*, 2000.

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *CVPR*, 2005.

[9] A. Delong, A. Osokin, H. Isack, and Y. Boykov. Fast approximate energy minimization with label costs. *IJCV*, 2011.

[10] A. Ellis and J. Ferryman. PETS2010 and PETS2009 evaluation of results using individual ground truthed single views. *AVSS*, 2010.

[11] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. *ECCV*, 2008.

[12] H. Isack and Y. Boykov. Energy-based geometric multi-model fitting. *IJCV*, 2011.

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

[14] Z. Khan, T. Balch, and F. Dellaert. MCMC data association and sparse factorization updating for real time multitarget tracking with merged and multiple measurements. *PAMI'06*.

[15] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *PAMI*, 28(10), 2006.

[16] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *PAMI*, 26(2), 2004.

[17] B. Leibe, N. Cornelis, K. Cornelis, and L. Van Gool. Dynamic 3D scene analysis from a moving vehicle. *CVPR'07*.

[18] B. Leibe, K. Schindler, N. Cornelis, and L. Van Gool. Coupled detection and tracking from static cameras and moving vehicles. *PAMI*, 30(10), 2008.

[19] S. Oh, S. Russell, and S. Sastry. Markov chain Monte Carlo data association for general multiple-target tracking problems. *IEEE Conference on Decision and Control*, 2004.

[20] K. Okuma, A. Taleghani, N. de Freitas, J. Little, and D. Lowe. A boosted particle filter: Multitarget detection and tracking. *ECCV*, 2004.

[21] D. B. Reid. An algorithm for tracking multiple targets. *IEEE T Automat Contr*, 24(6), 1979.

[22] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer. Optimizing binary MRFs via extended roof duality. *CVPR'07*.

[23] R. Stiefelhagen, K. Bernardin, R. Bowers, J. S. Garofolo, D. Mostefa, and P. Soundararajan. The CLEAR 2006 evaluation. *CLEAR*, 2006.

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

[25] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by Bayesian combination of edgelet part detectors. *IJCV*, 75(2), 2007.

[26] Z. Wu, T. H. Kunz, and M. Betke. Efficient track linking methods for track graphs using network-flow and set-cover techniques. *CVPR*, 2011.

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