Continuous Energy Minimization for Multi-Target Tracking

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

Abstract—Many recent advances in multiple target tracking aim at finding a (nearly) optimal set of trajectories within a temporal window. To handle the large space of possible trajectory hypotheses, it is typically reduced to a finite set by some form of data-driven or regular discretization. In this work we propose an alternative formulation of multi-target tracking as minimization of a continuous energy. Contrary to recent approaches, we focus on designing an energy that corresponds to a more complete representation of the problem, rather than one that is amenable to global optimization. Besides the image evidence, the energy function takes into account physical constraints, such as target dynamics, mutual exclusion, and track persistence. In addition, partial image evidence is handled with explicit occlusion reasoning, and different targets are disambiguated with an appearance model. To nevertheless find strong local minima of the proposed non-convex energy we construct a suitable optimization scheme that alternates between continuous conjugate gradient descent and discrete trans-dimensional jump moves. These moves, which are executed such that they always reduce the energy, allow the search to escape weak minima and explore a much larger portion of the search space of varying dimensionality. We demonstrate the validity of our approach with an extensive quantitative evaluation on several public datasets.

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

1 Introduction

Simultaneously keeping track of multiple targets in a video while preserving their identities remains a challenging task of computer vision. Accurate tracking is crucial for many applications, such as pedestrian safety, motion and scene analysis, and video surveillance. Despite enormous progress in recent years, the tracking abilities of humans still easily exceed state-of-the-art algorithms in real world scenarios, both in terms of precision and accuracy – if given enough time to process the data. Many recent approaches to tracking pursue a *tracking-by-detection* strategy, where the targets are detected in a pre-processing step, usually either by background subtraction or using a discriminative classifier, from which the trajectories are later estimated. The benefit is an improved robustness against drifting and the possibility of recovering from tracking failure. In the comparatively simple single-target setting, where only one object is present in the scene, tracking can be approached by searching for the object of interest within the expected area and forming a plausible trajectory by connecting the object's locations over time. When a higher, often unknown number of targets is observed simultaneously, the problem becomes much more complicated, because it is no longer obvious which object corresponds to which detection. This task of correctly identifying different objects over time is often referred to as *data association*. Moreover, motion, appearance, and visibility of objects are influenced by mutual dependencies that have to be taken into account. From a probabilistic point of view this entails inference – often maximum a-posteriori (MAP) – in a posterior distribution over several variables that are not independent. The resulting optimization problem is highly non-convex (in case of a continuous domain) or non-submodular (in the discrete case), and thus cannot be optimized globally without major simplifying assumptions.

Yet, several recent multi-target tracking formulations aim to obtain a (nearly) globally optimal set of trajectories within a temporal window [3], [7], [23], [24], [31], [34], [43]. In order to make (near) global optimization possible and efficient, the state space is reduced by restricting the possible target locations to a finite set and the energy function is simplified. While global optimality undoubtedly has many benefits, we must also not lose sight of the actual purpose of formulating multi-target tracking as an energy minimization problem: the energy should adequately reflect the task at hand so that low-energy solutions are close to the true situation. Unfortunately, in the realm of multi-target tracking typical specifications of the desirable aspects do not lead to models that can be globally optimized.

In this work we investigate the question whether it is really beneficial for multi-object tracking to (overly) restrict the energy function in order to guarantee global optimality. In contrast to previous work, we attempt to design the objective function such that it offers a more complete representation of the various aspects of the problem. Our energy is defined in continuous space.

- *A. Milan and S. Roth are with the Technische Universität Darmstadt, Dept. of Computer Science, Fraunhoferstr. 5, 64283 Darmstadt, Germany. E-mail: {anton.milan@gris, sroth@cs}.tu-darmstadt.de*
- *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*

The energy depends on the locations and motions of *all* targets in *all* frames, including cases where image evidence is missing, and explicitly includes physical constraints, such as smoothness of motion and mutual exclusion. It is beneficial to model these terms in the continuous domain, since they describe the true situation more closely than ones that operate in a discrete setting. The price to pay is having to forgo global optimality, since such a complex model of multi-target tracking is unlikely to be convex. Nevertheless, local optima of our energy yield better results in practice, both visually and in terms of quantitative evaluation with respect to ground truth.

To make the optimization efficient, all energy terms are formulated as functions that can be computed and differentiated in closed form. Hence, computationally efficient gradient-based optimization methods can be applied. To find strong local minima and to reduce the influence of the initialization, we run standard conjugate gradient descent from several starting points. Additionally, this purely continuous minimization is extended by a set of trans-dimensional jump moves, which enable the search to escape the initial basin of attraction and explore a larger region of the energy landscape. To support our hypothesis that accurate modeling might be more important than optimality guarantees for tracking performance, we run extensive experiments on various public datasets and show state-of-the-art results quantitatively measured by standard multi-target tracking metrics.

The main contribution of this paper is an energy-based model of multi-target tracking that

- is defined over all target locations (in continuous space) and all video frames in a given time window;
- includes per-frame detection evidence, appearance, dynamics, persistence, and collision avoidance;
- explicitly handles partial as well as full inter-object occlusion; and
- can be computed and differentiated efficiently in closed form.

Furthermore, we provide an empirical study on the influence of all major parameters of the model, and an analysis of various optimization strategies for model inference, ranging from greedy search to more randomized and sampling-based algorithms.

Parts of this work have appeared in [2], [4]. Here, we for the first time describe the complete model, and include an additional appearance component, an extended evaluation with more data sets, an empirical study of the contributions of individual model parts, and an evaluation of different optimization strategies.

2 Related Work

Tracking has been an active research area for many years and the amount of related literature is vast. Here, we concentrate on prior work in *visual multi-object tracking*.

Especially early on, many tracking algorithms utilized *recursive* methods, where the current state is estimated only using information from previous frames. Kalman filter approaches [9], [32] are a prominent example. Later, particle filtering (also known as sequential Monte Carlo) was introduced, where a set of weighted particles – sampled from a proposal distribution – is maintained to represent the current, hidden state [10], [30], [37]. This allows handling non-linear multi-modal distributions. However, as the number of targets grows, a reliable representation of the posterior requires an ever increasing amount of samples and is hard to handle in practice. Data association is usually approached by probabilistic filtering (JPDAF) [19] or by Markov chain Monte Carlo sampling techniques (MCMCDA) [29].

Over the past few years, *non-recursive* tracking methods have grown more and more popular [3]–[7], [11], [24], [27], [28], [41], [43]. The commonality of these methods is that all trajectories are estimated jointly within a given time window. However, to keep the computation tractable, the solution space is restricted to a finite number of states, which is usually done by only allowing trajectories to pass exactly through either non-maxima suppressed object detections [24], [27], [43], or through locations on a regular discrete grid [3], [7]. Leibe *et al*. [27] couple the tasks of object detection and trajectory estimation through a quadratic binary program, which is then solved to local optimality by custom heuristics. Jiang *et al*. [24] cast the task of tracking multiple targets as an integer linear program (ILP) with linear constraints to enforce that the layout between targets does not change in adjacent frames. The solution is then obtained by LP-relaxation, which cannot guarantee global optimality in general, but achieves it in most cases nonetheless. To allow objects to pass through occlusions, a special "occluded"-node is introduced. A drawback of this approach is that the number of targets needs to be known a priori, which is a serious limitation for many applications. Furthermore, due to the undefined locations of occluded targets, there is no chance to avoid collisions between them. To achieve high tracking accuracy and to obtain a plausible solution, a precise localization of occluded targets is crucial [21]. Berclaz *et al*. [7] divide the tracking area into a grid of disjoint cells and introduce a virtual location, which can spawn new trajectories and absorb existing ones at certain locations (*e.g*., doors or image borders). The solution of the resulting integer linear program can again be obtained by LP-relaxation, or using the K-shortest paths algorithm [8], which significantly speeds up computation. The framework can be further extended to include the object appearance [34], thereby reducing the number of identity switches between targets. While this approach achieves high-quality results, the recovered trajectories suffer from aliasing due to the discretization of the location space and appear unnatural, even when enriched with a dynamic model [3]. A network flow approach for global multi-target tracking was introduced by Zhang *et al*. [43]. Observation and transition edges between individual detections form a graph where their

capacity represents the likelihood of target presence and motion. An optimal set of trajectories without occlusion handling is found by a min-cost flow algorithm. Brendel *et al*. [11] divide the data association problem into disjoint subgraphs and solve each one independently. Using soft and hard constraints, the algorithm is guaranteed to converge. Henriques *et al*. [23] pursue a similar approach, but introduce merge and split events to go beyond one-to-one matches between graph nodes. The recent work of Benfold and Reid [5] presents highly accurate results computed in real time. Similar to [20], short tracklets are generated by robust feature point tracking. The final trajectories are then found by Monte Carlo sampling. Occlusions are avoided by choosing an elevated viewpoint and detecting the heads of pedestrians instead of full bodies. Head-based tracking in dense crowds is also employed by Rodriguez *et al*. [33], where the solution is obtained by minimizing a binary energy function with a constraining term to enforce the correct number of targets. While a high camera viewpoint can be assumed in many surveillance scenarios, this is generally not feasible in other applications (*e.g*., driver assistance or entertainment).

Occlusion reasoning plays an important role in many areas of computer vision, including pose estimation [14], [35], and object detection [15], [39], [40]. The reason why occlusion modeling improves results is consistent in all cases: the knowledge that the observed object is only partially (or not at all) visible predicts that less evidence will be found in the image, and the appraisal of the evidence can be adapted accordingly.

In the realm of multi-target tracking the inter-object occlusion problem has either been ignored [4], [7], [28], or handled iteratively [42], [43]. Xing *et al*. [42] generate short tracklets without occlusion reasoning and then connect tracklets to longer trajectories such that the connections can bridge gaps due to occlusions. Zhang *et al*. [43] address data association with a network flow approach, where an optimal subset of trajectories is greedily extended into occluded regions in a post-processing step. Wojek *et al*. [39] extend a full-body detector with six part detectors to enrich the space of target hypotheses. Each detection is then weighted by its expected visibility computed from a 3D scene model. Somewhat similar to our approach, Breitenstein *et al*. [10] increase the target likelihood if another target exists nearby. However, our occlusion reasoning provides an accurate approximation to the actual fraction of the target visibility.

Seriously crowded environments, where large numbers of dynamic targets and frequent occlusions make tracking difficult even for a human observer, are rarely processed at the level of individual targets. Notable exceptions include the work of Kratz and Nishino [25], which relies on spatio-temporal motion patterns of the crowd. Li *et al*. [28] also address crowded environments and learn tracklet associations online. Both approaches do not include any dedicated occlusion reasoning.

The proposed global occlusion model (*cf*. Sec. 3.3) is closely integrated into our continuous tracking framework, and can easily handle a large number of targets. Moreover, it is able to accurately estimate pairwise visibility dependencies between all targets.

Symbol	Description
$\mathbf{X}$	world coordinates of all targets in all frames
$\mathbf{X}_i^t$	(X,Y) world coordinates of target i in frame t
$\mathbf{x}_i^t$	(x,y) image coordinates of target i in frame t
F, N	total number of frames and targets, respectively
$F(i)$	number of frames where target i is present
s_i, e_i	first, respectively last frame of trajectory i
$N(t), D(t)$	number of targets, respectively detections in frame t
$\mathbf{D}_g^t$	(X,Y) world coordinates of detection g in frame t

TABLE 1. Notation.

3 Multi-Target Tracking

3.1 Preliminaries and notation

To ease understanding, we first introduce the general structure and notation used throughout the paper. The state vector $\mathbf{X}$ consists of the (X, Y) world coordinates of all N targets in a sequence of F frames. We assume that all targets move on a common ground plane, *i.e*. $Z = 0$. Note that the continuous location $\mathbf{X}_i^t \in \mathbb{R}^2$ of target i at time t is exactly defined for all frames $t \in \{s_i, \ldots, e_i\}$ within the temporal life span of the trajectory, even if the target is not associated with any detection or is entirely occluded. The temporal length of trajectory i is denoted $F(i) := e_i - s_i + 1$, where s_i and e_i, respectively, are the first and final frames. Our formulation does not assume the number of targets to be known a priori; this number may in fact vary from frame to frame. We thus denote the number of targets in frame t as $N(t)$. Similarly, $D(t)$ indicates the number of detections in frame t. The location of detection g in frame t itself is denoted as $\mathbf{D}_g^t$. Lower case letters $\mathbf{x}, x, y$ describe image coordinates. The notation is summarized in Table 1.

3.2 Continuous energy

Energy minimization methods have – in one form or another – become quite popular for multi-target tracking [7], [27], [43]. Their common objective is to set up a function that assigns every possible solution a cost (the "energy") and then (approximately) find the state with the lowest cost. An energy function for a certain application can be defined in many ways. In computer vision one often faces two major problems: (1) The input data is noisy and requires robust models; (2) an accurate representation that captures all relevant nuances of the real situation quickly becomes very complex. Together these two issues tend to produce complicated and highly non-convex objective functions (*cf*. Sec. 4). One is thus faced with a dilemma: Should the energy function be simplified until it is easily optimizable, or should it rather have the power to capture the complex situation, at the cost of less graceful mathematical properties? In the present work, we investigate the latter alternative for

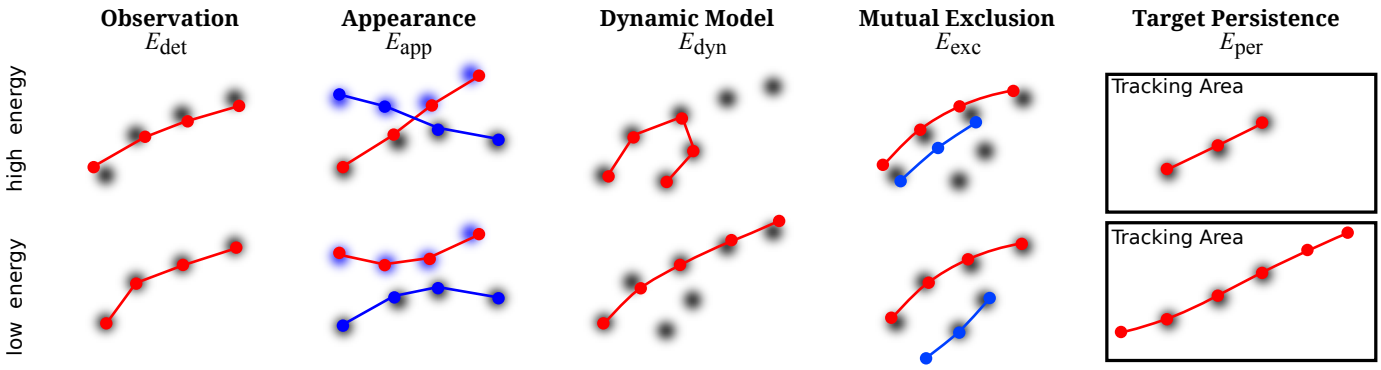

Fig. 1. The effects of different components of the energy function. The top row shows a configuration with a higher, the bottom row with a lower value for each individual term. The dark, smooth blobs denote detection locations. Different colors of the target locations (marked with circles) suggest distinguishable appearance between targets.

the case of tracking multiple objects in video. The energy we propose has been developed with an emphasis on precisely describing multi-object tracking. Algorithmic considerations were limited to keeping the function differentiable in closed form and thus efficient for gradient-based optimization. It turns out that for the case of multi-target tracking such an approach is rather successful.

Our energy function is a linear combination of six individual terms:

$$E = E_{\text{det}} + \alpha E_{\text{app}} + \beta E_{\text{dyn}} + \gamma E_{\text{exc}} + \delta E_{\text{per}} + \epsilon E_{\text{reg}}. \quad (1)$$

The data term E_{det} keeps the solution close to the observations; the term E_{app} captures the appearance of different objects to disambiguate data association; the three priors $E_{\text{dyn}}, E_{\text{exc}}$ and E_{per} promote plausible motion and enforce physical constraints; the regularizer E_{reg} keeps the solution simple and prevents over-fitting. The aim is then to find the state $\mathbf{X}^*$ that minimizes the high dimensional continuous energy from Eq. (1):

$$\mathbf{X}^* = \arg\min_{\mathbf{X}\in\mathbb{R}^d} E(\mathbf{X}). \quad (2)$$

Depending on the length of the sequence and the number of targets, the dimension of the search space d normally takes on values between 10^3 and 10^4. In the remainder of this section we explain each individual term and its functionality in more depth. Fig. 1 illustrates the first five components. The influence of the individual terms is examined in Sec. 6.1 by adjusting their respective weights or discarding them entirely.

3.2.1 Observation model

In this work we concentrate on people as tracking targets, and follow the well established tracking-by-detection approach. Likely pedestrian locations are found with a sliding-window linear SVM detector. The features employed in the detector include histograms of oriented gradients (HOG) [13] and histograms of relative optic flow (HOF) [38]. Detection peaks are found by non-maxima suppression (NMS) and projected onto the ground plane of the world coordinate system, where they form the image evidence for tracking. We limit ourselves to using non-maxima suppressed detections to reduce the computational cost, but note that this is not a major limitation of our approach; it could easily be extended to use a per-pixel target likelihood instead (*cf*. [10]). The intrinsic and extrinsic camera parameters required for the projection are constant for static cameras and can be inferred by structure-from-motion for moving cameras (as done, *e.g.*, in [16] for multi-target tracking). Hence, the requirement of a calibrated camera does not pose a major limitation and enables more accurate modeling of target dynamics and interaction.

The main purpose of the data term is to keep the trajectories close to the observations. In other words, the energy should be minimal when the location of each target precisely matches a detection. To capture the localization uncertainty of the object detector, the energy smoothly increases with the distance between the estimated object location $\mathbf{X}_i^t$ and a detection location $\mathbf{D}_g^t$. This behavior is modeled by an isotropic (inverse) bell-shaped function centered at the detector output,

$$E^*_{\text{det}}(\mathbf{X}) = \sum_{i=1}^{N}\sum_{t=s_i}^{e_i}\left[\lambda - \sum_{g=1}^{D(t)} \omega_g^t \frac{s^2}{\|\mathbf{X}_i^t - \mathbf{D}_g^t\|^2 + s^2}\right]. \quad (3)$$

Each detection is weighted by its confidence ω and the scalar s accounts for the object size, *i.e.* the area on the ground plane occupied by that object. It is set to 35cm for pedestrian tracking. The offset λ is added uniformly to all existing targets to penalize all those with no image evidence. This penalty, however, must not be applied if a target is occluded and consequently cannot possibly be "seen" by the detector. It is therefore scaled by the fraction of the visibility v_i^t of that target:

$$E_{\text{det}}(\mathbf{X}) = \sum_{i=1}^{N}\sum_{t=s_i}^{e_i}\left[v_i^t \cdot \lambda - \sum_{g=1}^{D(t)} \omega_g^t \frac{s_g^2}{\|\mathbf{X}_i^t - \mathbf{D}_g^t\|^2 + s_g^2}\right]. \quad (4)$$

The global occlusion reasoning including the computation of v is explained in detail in Sec. 3.3. We also defer the discussion of the appearance term to Sec. 3.4, as it relies on the visibility fraction of individual targets.

3.2.2 Dynamic model

A defining property of tracking (as opposed to independent object detection per frame) is that objects move slowly relative to the frame rate, and in most cases also smoothly. This gives rise to constraints on the target motion, captured by a dynamic model. A simple constant velocity model that minimizes the distance between consecutive velocity vectors is powerful enough to capture the motion of objects in many real scenarios:

$$E_{\text{dyn}}(\mathbf{X}) = \sum_{i=1}^{N} \sum_{t=s_i}^{e_i-2} \left\| \mathbf{X}_i^t - 2\mathbf{X}_i^{t+1} + \mathbf{X}_i^{t+2} \right\|^2. \quad (5)$$

On one hand, the dynamic model helps reduce identity switches by favoring straight paths. On the other hand, the detections are often misaligned and do not form smooth curves. Naive smoothing might yield visually pleasing results, but is not appropriate to achieve high data fidelity and thus high tracking precision. The dynamic model as part of a global energy function can be seen as a form of "intelligent smoothing", yielding trajectories that are natural and smooth, while at the same time avoiding collisions and not drifting too far away from the actual observations.

3.2.3 Mutual exclusion

Collision avoidance is a crucial aspect when tracking multiple targets (*cf*. Sec. 6.1 and Fig. 9). In our model a continuous penalty is applied to configurations in which two targets come too close to each other:

$$E_{\text{exc}}(\mathbf{X}) = \sum_{t=1}^{F} \sum_{i,j \neq i}^{N(t)} \frac{s}{\|\mathbf{X}_i^t - \mathbf{X}_j^t\|^2}. \quad (6)$$

Note that the penalty is closely related to the intersection of the target volumes, which is also used by some authors [16], but our variant goes to infinity in the impossible case when both objects occupy the same 3D space. Besides enforcing the obvious physical constraint, a mutual exclusion term also ensures that one piece of image evidence can be explained by at most one target. This is especially important when dealing with soft observation models that exhibit a smooth falloff around the detection (*i.e.*, target locations are not clamped to the exact location of the detector output), since otherwise the same peak could give rise to multiple trajectories.

Our formulation of the exclusion model can handle two notoriously difficult problems: On one hand, the pairwise distance between all targets is taken into account at all frames. Hence, two targets cannot occupy the same space, even if both are occluded. On the other hand, if one detection of two neighboring targets is missing, the targets will be pushed apart just as much as needed to avoid a physically impossible situation. Tracking on a discrete grid does not allow intermediate steps and the entire trajectory may be discarded.

Note that our approach does not perform an explicit assignment between target hypotheses and measurements (detections). Data association is indirectly achieved, mainly by two continuous terms – observation and mutual exclusion. Such soft assignments not only produce visually more pleasing and physically more plausible trajectories, but also offer a more flexible interpretation of the data due to the continuous state space.

3.2.4 Trajectory persistence

Missing evidence can lead to track fragmentation or abrupt track termination in the middle of the tracking area. To encourage trajectories to start and end along image borders or along a predefined perimeter, tracks that do not obey this requirement are penalized. To keep the term both robust and smooth, we use a sigmoid centered on the border of the tracking area:

$$E_{\text{per}}(\mathbf{X}) = \sum_{\substack{i=1,\dots,N \\ t \in \{s_i, e_i\}}} \frac{1}{1 + \exp\big(-q \cdot b(\mathbf{X}_i^t) + 1\big)}, \quad (7)$$

where $b(\mathbf{X}_i^{s_i})$ and $b(\mathbf{X}_i^{e_i})$ measure the distance of the first, respectively last known location of target i to the closest border of the tracking area and the parameter q represents the soft entry margin and is set to $q = 1/s$, where $s = 35$cm is the target size as before.

3.2.5 Regularizer

Finally, a regularizer is needed to prevent the number of targets from growing arbitrarily large so as to better fit the data. To that end, we simply penalize the number of existing targets. It turns out that including the trajectory length into the regularization term leads to better performance, because solutions with many short tracks are less likely. These two terms are combined to form

$$E_{\text{reg}}(\mathbf{X}) = N + \sum_{i=1}^{N} \frac{1}{F(i)}. \quad (8)$$

Note that the second term can be weighted individually to adjust the importance of the lengths of the trajectories. Although empirically this leads to slightly better performance on some test sequences, we prefer to set it uniformly to 1 in all our experiments. Having fewer parameters facilitates the search for a good parameter set and avoids over-fitting.

3.3 Global occlusion reasoning

Having introduced our basic tracking framework, we now turn to our explicit occlusion reasoning scheme. In typical real-world scenarios three different types of occlusion take place: *(1)* in crowded scenes, targets frequently occlude each other causing *inter-object occlusion*; *(2)* a target may move behind static objects like trees, pillars, or road signs, which are all examples of common *scene occluders*; *(3)* depending on the object type, extensive articulations, deformations, or orientation changes may cause *self-occlusion*. All three types of occlusion reduce – or completely suppress – the image evidence for a target's presence, and consequently incur penalties

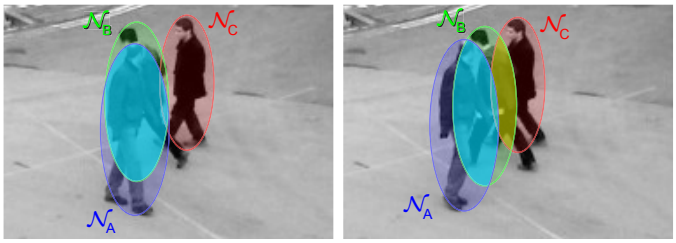

Fig. 2. A typical example of inter-object occlusion. In our proposed occlusion model targets are represented as anisotropic Gaussians in image space (red, green, blue), whereas pairwise occlusions between all targets (cyan, yellow) are approximated by products of Gaussians.

in the observation model. Specifically, in our tracking-by-detection setting they cause the object detector to fail and thereby increase E_{det}. However, simply treating occlusion as missing data, *i.e. ignoring the fact that the observed occluder actually predicts the lack of evidence, can seriously impair tracking performance*.

Consequently, explicit occlusion handling is important for successful multi-target tracking. Unfortunately, principled modeling of occlusion dependencies is rather tricky as the following example illustrates (see Fig. 2):

> *If target* A *is at location* $\mathbf{X}_{\text{A}}$*, then target* B *at* $\mathbf{X}_{\text{B}}$ *is occluded; but if* A *is a bit further to the left and* B *slightly further to the right, then* B *is partially visible; however then it would partially occlude target* C*;* etc.

In order to deal with situations where dynamic targets occlude each other, the main task is to overcome the difficulties that arise from the complex dependence between a target's visibility and the trajectories of several other targets, which could potentially block the line of sight. An explicit occlusion model thus leads to complicated objective functions, which tend to be difficult and inefficient to optimize. Therefore, most previous approaches either ignore the issue altogether, or resort to some form of greedy heuristic, usually separating target localization from occlusion reasoning.

We present a method that tightly couples both trajectory estimation and explicit inter-object occlusion reasoning. Note that it can be trivially extended to handle scene occluders. Not surprisingly, taking into account occlusions directly during trajectory estimation significantly reduces the number of missed targets and lost tracks – especially in highly crowded environments.

3.3.1 Analytical global occlusion model

Our approach handles mutual occlusion between all targets with a closed-form, continuously differentiable formulation. Since this procedure is identical for each frame, the superscript t is omitted for better readability.

Relative overlap. Let us for now assume that each target i is associated with a binary indicator function

$$\mathbf{o}_i(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} \in \mathcal{B}(\mathbf{X}_i) \\ 0, & \text{otherwise,} \end{cases} \tag{9}$$

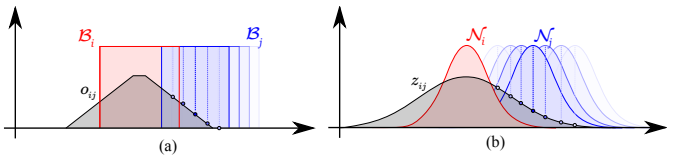

Fig. 3. Schematic illustration (in 1D) of targets' overlap as a function of the occluder's position. In case of bounding boxes *(a)*, the overlap o_{ij} is non-differentiable on the borders. In contrast, our occlusion term z_{ij} is a Gaussian.

which is 1 on the bounding box $\mathcal{B}(\mathbf{X}_i)$ of target i. The total image area of target i is thus given as $\int \mathbf{o}_i(\mathbf{x})\, d\mathbf{x}$. To compute the relative area of target i that is occluded by target j, we simply have to calculate the (normalized) integral of the product of both indicator functions:

$$\frac{1}{\int \mathbf{o}_i(\mathbf{x})\, d\mathbf{x}} \int \mathbf{o}_i(\mathbf{x}) \mathbf{o}_j(\mathbf{x})\, d\mathbf{x} \tag{10}$$

Note that we assume here that target j is in front of target i; we will address the more general case below. If we define the target visibility using the relative area as given in Eq. (10), then the visibility is not differentiable w.r.t. the object positions of $\mathbf{X}_i$ or $\mathbf{X}_j$, which precludes gradient-based optimization methods (*cf*. Fig. 3(a)).

To address this issue we here propose to use a Gaussian "indicator" function $\mathcal{N}_i(\mathbf{x}) := \mathcal{N}(\mathbf{x}; \mathbf{c}_i, \mathbf{C}_i)$. Besides achieving differentiability, this is motivated by the fact that a Gaussian blob is a crude, but reasonable approximation for the shapes of many objects (see Fig. 2 for an illustration). In our case, each person in image space is represented by an anisotropic Gaussian with $\mathbf{c}_i = \mathbf{x}_i$ and

$$\mathbf{C}_i = \begin{pmatrix} \frac{1}{2}\left(\frac{s_i}{2}\right)^2 & 0 \\ 0 & \left(\frac{s_i}{2}\right)^2 \end{pmatrix}$$

with s_i being the target's height on the image plane. As before, we compute the area of overlap by integrating the product of the two "indicator" functions, here Gaussians:

$$z_{ij} = \int \mathcal{N}_i(\mathbf{x}) \cdot \mathcal{N}_j(\mathbf{x})\, d\mathbf{x} \tag{11}$$

Besides differentiability, the choice of Gaussians allows this integral to be computed in closed form. Conveniently, the integral is another Gaussian [12]: $z_{ij} = \mathcal{N}(\mathbf{c}_i; \mathbf{c}_j, \mathbf{C}_{ij})$ with $\mathbf{C}_{ij} = \mathbf{C}_i + \mathbf{C}_j$ (see Fig. 3 for a schematic illustration). Since we are interested in the *relative* overlap that corresponds to the fraction of occlusion between two targets, we compute it using the unnormalized Gaussian

$$V_{ij} = \exp\left(-\tfrac{1}{2}[\mathbf{c}_i - \mathbf{c}_j]^\top \mathbf{C}_{ij}^{-1} [\mathbf{c}_i - \mathbf{c}_j]\right), \tag{12}$$

which is differentiable w.r.t. $\mathbf{c}_i$ and $\mathbf{c}_j$ and has the desired property that $V_{ij} = 1$ when $\mathbf{c}_i = \mathbf{c}_j$. Moreover, due to the symmetry of Gaussians we have $V_{ij} = V_{ji}$.

Depth ordering. To also take into account the depth ordering of potentially overlapping targets, we could make use of a binary indicator variable, which once

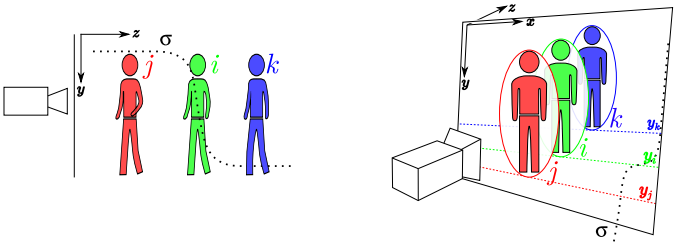

Fig. 4. Target i has a non-zero overlap with j and with k. However, it is only occluded by j. Hence, the overlap is weighted with a sigmoid σ (dotted line) centered on y_i.

Fig. 5. Instead of extracting and comparing full bounding boxes *(b)*, we propose to weigh the area using anisotropic Gaussians *(c)*. The energy remains differentiable, and the influence of undesired background pixels is reduced.

again has the issue of making the energy function non-differentiable. We again replace it with a continuous, differentiable version and use a sigmoid along the vertical image dimension centered on y_i (*cf*. Fig. 4): $\sigma_{ij} = (1 + \exp(y_i - y_j))^{-1}$. Note that this definition assumes a common ground plane as well as a camera at a rather low viewpoint and in standard landscape or portrait orientation, such that the depth order corresponds to the order of the targets' y-coordinates (it is however straightforward to extend the idea to more general setups). Also note that if we assume small variation in target size, then the occluder will always appear larger than the occluded object on the image plane and hence will entirely cover the farther target if their center points coincide.

Visibility. To define the overall visibility of each target, we first define an occlusion matrix $\mathcal{O} = (\mathcal{O}_{ij})_{i,j}$ with $\mathcal{O}_{ij} = \sigma_{ij} \cdot V_{ij}, i \neq j$ and $\mathcal{O}_{ii} = 0$. The entry in row i and column j of $\mathcal{O}$ thus corresponds to (a differentiable approximation of) the fraction of i that is occluded by j. Disregarding cases where multiple occluders cover the same limited fraction of a target, we can now approximate the total occlusion of i as $\sum_j \mathcal{O}_{ij}$. A straightforward definition of the visible fraction of i would thus be $\max(0, 1 - \sum_j \mathcal{O}_{ij})$. However, to avoid the non-differentiable $\max$ function, we prefer an exponential function and define the visibility for target i as

$$v_i(\mathbf{X}) = \exp\left(-\sum_j \mathcal{O}_{ij}\right). \tag{13}$$

This definition allows us to efficiently approximate the visible area by taking into account mutual occlusion for each pair of targets. Furthermore, by consistently using appropriate differentiable functions the entire energy has a closed form and remains continuously differentiable.

Limitations. The main limitation of this approach is that targets are represented with a simple oval shape. However, our experiments show that the actual fraction of occlusion can be estimated quite reliably even for pedestrians, despite their non-rigid, articulated motion.

3.4 Appearance model

The appearance of an object may provide important cues for disambiguating it from the background and from other objects. This aspect has previously either been ignored [2], [4], [7], or addressed only in the discrete setting [26], [34], [43]. Here, we present a novel appearance term that is continuously differentiable in closed form, thus admitting gradient-based optimization.

Assuming that an object's color remains constant over time and that lighting changes slowly, our appearance model imposes a higher penalty in cases of abrupt changes. To maintain the benefits of the continuous formulation, it is desirable to describe the appearance of an object analytically. To ensure that the energy remains smooth without costly interpolation, we propose to use Gaussian weighted regions (*cf*. Fig. 5). This not only ensures differentiability, but a closed-form gradient. This is also motivated by the fact that the object of interest typically occupies the central area inside the bounding box. The background pixels along the borders and in the corners are therefore naturally downweighted, while the pixels closer to the center receive higher weights.

Formally, the Gaussian weighted histogram count of the image region occupied by target i in frame t is defined as

$$h_n(\bar{\mathbf{x}}_i^t) = \sum_{\mathbf{x}} \left[\mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}_i^t, \mathbf{\Sigma}_i^t) \cdot H_n(\mathbf{x})\right], \tag{14}$$

where H_n is a binning function

$$H_n(\mathbf{x}) = \begin{cases} 1, & \text{if } I(\mathbf{x}) \text{ falls into bin } n \\ 0, & \text{otherwise,} \end{cases} \tag{15}$$

and $\bar{\mathbf{x}}$ is the center of the target's bounding box. We employ the widely used Bhattacharyya coefficient

$$BC(\mathbf{X}_i^t) := \sum_n^{\#\text{ bins}} \sqrt{h_n(\bar{\mathbf{x}}_i^t) * h_n(\bar{\mathbf{x}}_i^{t+1})}. \tag{16}$$

for histogram comparison. In our experiments a standard RGB color histogram with 16 bins per channel yields the best results. Obviously, the appearance of a target will change if it becomes occluded and thus should not be taken into account. We therefore reduce the influence of the appearance term in such cases by weighting the histogram deviation with the geometric mean of the visibilities (*cf*. Sec. 3.3) of the two bounding boxes:

$$AC(\mathbf{X}_i^t) = \bar{v}_i^t(\mathbf{X}) \cdot (1 - BC(\mathbf{X}_i^t)) \tag{17}$$

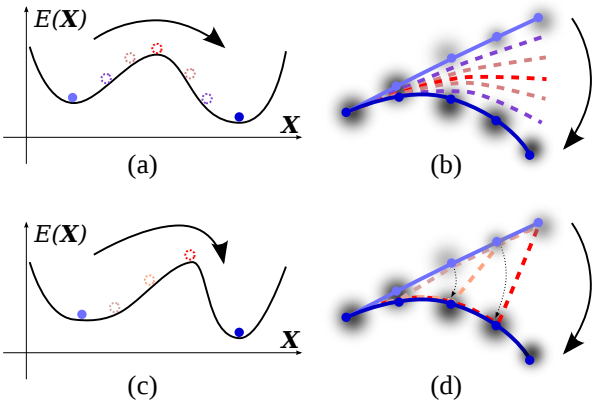

Fig. 6. Illustration of the non-convexity of the continuous tracking formulation. To get from the light blue solution (weaker optimum) to the dark blue one (stronger optimum) in a continuous state space one has to overcome a ridge of high energy. *(a,b)* Keeping E_{dyn} low incurs high penalties in E_{obs} as one moves away from the observations. *(c,d)* Keeping E_{obs} low incurs high penalties in E_{dyn} as the paths gets distorted to fit the observations. With a peaked likelihood intermediate cases are even worse.

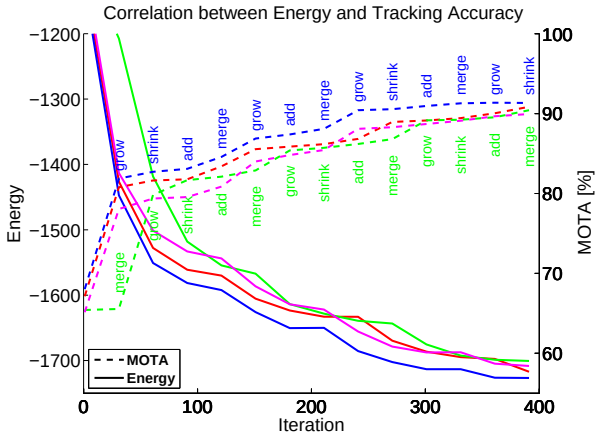

Fig. 7. Four optimization runs started from four different initializations on the sequence *S2L1*. The proposed energy (solid) correlates well with tracking performance w.r.t. ground truth (dashed). Energy values have been scaled uniformly for ease of visualization.

with

$$\bar{v}_i^t(\mathbf{X}) := \sqrt{\left(v_i^t(\mathbf{X}) \cdot v_i^{t+1}(\mathbf{X})\right)}. \tag{18}$$

Instead of simply adding this penalty to the energy, we found it to be beneficial in practice to use a soft threshold to better discriminate between true matches with a high similarity, *i.e.* low energy value, and identity switches. To that end, the final appearance term uses a sigmoid:

$$E_{\text{app}}(\mathbf{X}) = \sum_{i=1}^{N} \sum_{t=s_i}^{e_i-1} \frac{1}{1 + \exp(a_1 - a_2 * AC(\mathbf{X}_i^t))}. \tag{19}$$

The parameters $a_1 \approx 7.2, a_2 \approx 33.7$ are determined by a least squares fit to a subset of the available data.

Our appearance model is designed to fit gradient based optimization methods. As we show in Sec. 6, including appearance significantly reduces the number of identity switches and track fragmentations, though not increasing the average accuracy on the chosen datasets. Moreover, it forces the tracker to follow the targets more closely thereby increasing the tracking precision. We believe that appearance will be even more helpful in high resolution videos – where targets usually provide more color information – or in situations with stronger appearance variation than in existing benchmarks.

4 Optimization

The energy in Eq. (1) described in Sec. 3.2 is clearly not convex. In fact, it is not unlikely that a realistic model of multi-target tracking cannot be convex: It is easy to construct examples that have two virtually equal minima separated by a ridge of high energy (*cf*. Fig. 6). The main reason for this behavior is the high-order dependence between variables caused by physical constraints.

To minimize the energy function in Eq. (1) locally, we use the standard conjugate gradient method. Upon convergence, a jump move is executed (unless it would increase the energy), which may change the dimensionality of the model. The jumps give the optimization a high degree of flexibility – the initial solution need not even have the correct number of targets. To speed up the optimization process, all trajectories are given the chance to execute a certain jump at the same time. Based on our experience, the order in which the jumps are executed is not crucial, because the optimization may choose to perform an inverse move to find the way towards a lower energy. Please refer to Sec. 6.2 for an empirical study on various optimization strategies.

Our data-driven strategy for changing the dimension of the state vector is reminiscent of reversible jump Markov Chain Monte Carlo methods [22], which has been applied to multi-target tracking in various ways [5], [20], [41]. A crucial difference to traditional Monte Carlo sampling is that our method is deterministic: It exploits the advantages of gradient descent over sampling within one mode, and performs jumps according to a prescribed schedule, only if they decrease the energy. The energy minimization algorithm is summarized in Alg. 1.

4.1 Transdimensional jumps

To escape weak local minima we introduce six types of jump moves, which change the configuration of the current solution, thereby altering the dimension of the current state $\mathbf{X}_{\text{curr}}$. By jumping to different regions of the search space while always lowering the energy, the optimization is able to find much stronger local minima. An example of an optimization run with jumps is shown in Fig. 8. Here, a weak trajectory (black) is removed entirely while a new one (green) is initialized. Note that each jump leads to a configuration with a lower energy.

Growing and shrinking. The time span during which a target is visible in the target area can be changed

by growing or shrinking its trajectory. To extend the trajectory's length, it is simply linearly extrapolated in space-time (both forward and backward).

Let $\mathbf{X}_i = \mathbf{X}_i^{s_i:e_i}$ denote the current state of the i^{th} trajectory defined between frames s_i and e_i. To evaluate the energy $E_{\text{new}} = E(\mathbf{X}_{\text{new}})$, the trajectory is extrapolated backwards for t frames resulting in

$$\tilde{\mathbf{X}}_i = (\mathbf{X}_i^{s_i-t:s_i-1}, \mathbf{X}_i) \tag{20}$$

leading to the new state

$$\mathbf{X}_{\text{new}} = \Big(\bigcup_{\substack{j=1\dots N \\ j\neq i}} \mathbf{X}_j \Big) \cup \tilde{\mathbf{X}}_i. \tag{21}$$

The procedure for forward extrapolation is analogous with $\tilde{\mathbf{X}}_i = (\mathbf{X}_i, \mathbf{X}_i^{e_i+1:e_i+t})$. Shortening is achieved by simply discarding t past or future positions of a target: $\tilde{\mathbf{X}}_i = \mathbf{X}_i^{s_i+t:e_i}$, respectively $\tilde{\mathbf{X}}_i = \mathbf{X}_i^{s_i:e_i-t}$. Such growing and shrinking steps help to pick up lost tracks and weed out spurious trajectories.

Merging and splitting of trajectories can effectively improve data association, *i.e.* eliminate identity switches and track fragmentations. Splitting at time t is implemented by breaking a path $\mathbf{X}_k$ in two:

$$\tilde{\mathbf{X}}_i = \mathbf{X}_k^{s_k:t}, \quad \tilde{\mathbf{X}}_j = \mathbf{X}_k^{t+1:e_k} \tag{22}$$

if the split yields lower energy. Merging is executed if two paths can be smoothly connected into one with lower energy, preserving physically plausible motion:

$$\tilde{\mathbf{X}}_k = (\mathbf{X}_i, \mathbf{X}_{\text{con}}^{e_i+1:s_j-1}, \mathbf{X}_j), \tag{23}$$

where $\mathbf{X}_{\text{con}}$ smoothly connects $\mathbf{X}_i^{e_i}$ and $\mathbf{X}_j^{s_j}$. Especially merging is a powerful tool to overcome temporary tracker failure due to weak evidence or occlusion.

Adding and removing. New trajectories can be generated at locations with strong detections, which are not yet assigned to any trajectory. The newly inserted tracks are started conservatively with three consecutive frames, $\tilde{\mathbf{X}}_i^{t-1:t+1} = (\mathbf{D}_g^t, \mathbf{D}_g^t, \mathbf{D}_g^t)$, but can grow or merge with existing ones at a later stage. An entire trajectory is removed if its total contribution to the energy is above a certain threshold, meaning that it reduces the overall likelihood of the current state, rather than increasing it. Adding helps to find missing trajectories not picked by the original tracking solution, whereas removal discards trajectories which have been pushed to a state with little evidence, unreasonable dynamics, and/or overlap with other trajectories.

We repeatedly iterate through the six different move types in a fixed, prescribed order (see Alg. 1). For each move type, the move parameters – *e.g.* the number of frames a trajectory is grown or the time step at which a trajectory is split – are optimized independently for each trajectory in a greedy fashion. It is important to note, however, that the optimization is not entirely greedy, since the move type order is fixed; thus it is not guaranteed that every step leads to the largest energy decrease. Please see Sec. 6.2 for a study on various optimization techniques.

Algorithm 1 Energy Minimization

Input: S initial solutions, detections D
Output: Best of $\leq S$ solutions
for $s = 1 \rightarrow S$ **do**
  while $\neg$ converged **do**
    for $m \in$ {grow, shrink, add, remove, merge, split} **do**
      for $i \in 1, \dots, N$ **do**
        `try jump move` m `on trajectory` i
        `(greedy parameter selection)`
        if $E^{\text{new}}(\mathbf{X}_s) < E^{\text{old}}(\mathbf{X}_s)$ **then**
          `perform jump move` m
        end if
      end for
      `perform conjugate gradient descent`
    end for
  end while
end for
Return: $\arg\min_{\mathbf{X}_s} E(\mathbf{X}_s)$

4.2 Initialization

Like any non-convex optimization, the result depends on the initial value from which the iteration is started. However, the described exploration strategy greatly weakens this dependency compared to a pure gradient method. By allowing jumps to low-energy regions of the search space, even if they are far away from the current state, the attraction to local minima is reduced: the weaker a minimum is, the more likely it gets to find a jump out of its basin of attraction that lowers the energy.

Empirically, even a trivial initialization with no targets works reasonably well, however takes many iterations to converge. We propose instead to use the output of an arbitrary simpler tracker as a more qualified initial value. In our experiments we used per-target extended Kalman filters (EKFs) where the data association is performed in a greedy manner using a maximum overlap criterion. Note that the EKF trackers are not intended to achieve the best possible performance, but rather to quickly generate a variety of starting values. This is accomplished by running the trackers several times with different parameters. Usually, different starting values converge to similar, albeit not identical solutions (see Fig. 7).

5 Implementation

Before presenting the experiments we would like to point out some implementation details.

Tracking area. In order to compute the distance to valid entry and exit points to enforce persistent trajectories (*cf*. Eq. 7), the boundary of the tracking area needs to be known. For our purposes we define a rectangular area on the ground so as to facilitate the computation of the distances. Targets outside its limits are excluded from the solution. This is, however, not a major limitation because

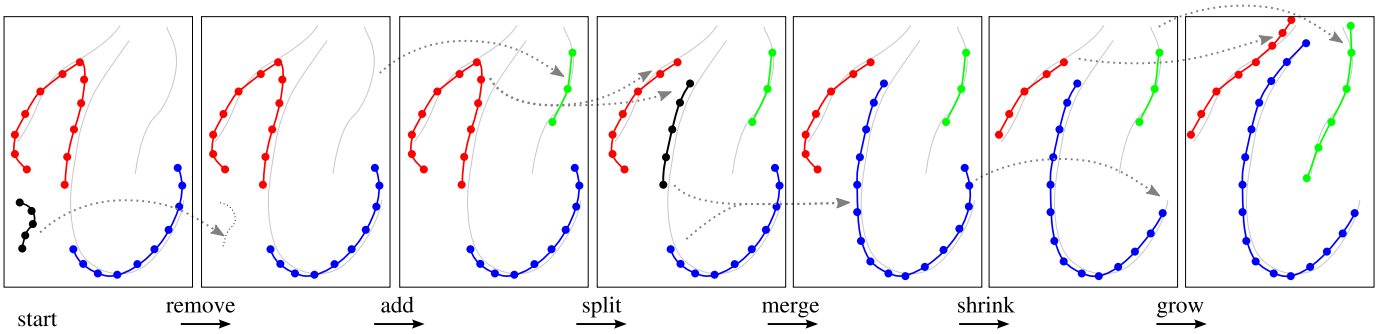

Fig. 8. The proposed jump moves make continuous optimization more flexible, allowing a variable number of targets. Even a poor initial configuration can be used to recover the true trajectories. The ground truth is rendered in gray.

the quadrilateral formed by the forward-projected image borders can easily be used instead as tracking area.

Run time. Given the detections, our current MATLAB/MEX implementation takes approximately 1s/frame to obtain one solution using explicit occlusion reasoning. Without the expensive occlusion computation, the optimization runs an order of magnitude faster, achieving near real-time performance. Unfortunately, computing color information and its derivatives for all pixels significantly slows down the optimization. While this can still be improved, the use of the appearance term is thus only recommended if computation time does not play a crucial role.

Convergence. As stated in Alg. 1, the energy is minimized until there is no jump that leads to a lower energy. Convergence is usually reached quickly (after 5 to 10 iterations). We set a maximum of 15 iterations because of timing constraints. Note that in some cases the results may still improve with more computational resources.

Parameters. Although the precise parameter values are highly dependent on the implementation at hand, we state them here for completeness. The weights α through ϵ are set to $\{.1, .02, .5, .7, .7\}$ and λ=.1 in all our experiments including the appearance term. Turning it off (*i.e.* setting $\alpha = 0$) also requires both δ and ϵ to be decreased slightly to a value of .6 to achieve best results. Finally, the setting for the basic energy without occlusion handling works robustly with parameters $\beta = .03, \gamma = \delta = \epsilon = .6, \lambda = .075$. Note that these parameter settings have been chosen conservatively and are not necessarily optimal for any particular dataset (*cf*. Fig. 9).

Our complete implementation together with all the necessary additional data, including detector output and ground truth, is available from the authors' website.

6 Experiments

In Sec. 3 we introduced an energy function that has been conceived with the primary goal of accurately reflecting the actual behavior of multiple interacting targets (*cf*. Fig. 7). As a consequence, the energy minimization can only be solved to local optimality, and there are no theoretical guarantees about the goodness of the solution. Our claim is that minimizing this function will nevertheless on average yield higher tracking accuracy. To empirically support this claim we performed an extensive experimental evaluation on various datasets.

Before presenting detailed quantitative results, we first analyze our approach in two regards: First, we examine the influence of the individual energy terms on the tracking performance and the robustness of the chosen parameters to variations of their respective values. Next, we compare different optimization strategies and their influence on the convergence rate and the final result.

6.1 Parameter study

Ideally, model parameters should be learned from example data, however that would require a large amount of annotated ground truth. We thus had to resort to determining the model parameters manually, which is not only tedious, but carries the danger of over-fitting. To mitigate this, we use only one parameter set per method for all test sequences, even though they exhibit strong variations both visually and in terms of target behavior.

To examine the influence of each individual weight of the energy in Eq. (1), we run our tracking algorithm and modify the corresponding parameter while keeping all the other ones fixed. In Fig. 9, for each term, the relative change in performance, as measured by MOTA, is plotted against the parameter value. For illustration, the average mean-normalized value is shown along with error bars, depicting the variation between various sequences. Note that even a relatively drastic scaling of the weights (*e.g.*, by a factor of $1/2$ or 2) hardly affects the overall performance. The strongest decline can be observed when γ – the weight for target exclusion – is set too low. This once again demonstrates the importance of explicitly modeling the spatial dependencies to avoid situations with overlapping targets. Moreover, we can conclude that the results are stable over a range of settings and tracking performance is only slightly affected by parameter changes within a reasonable range.

6.2 Optimization strategies

There are many possible ways of integrating discontinuous jump moves into an optimization scheme. To

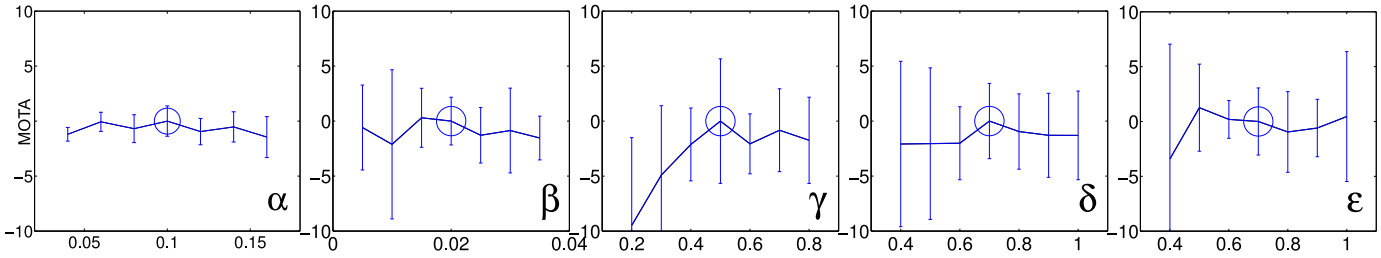

Fig. 9. Influence of individual parameters on tracking performance. Each plot shows the relative change in performance (measured by MOTA) by changing the weight of a single energy component while keeping the other ones fixed. The results shown here are averaged over all datasets and normalized for better readability. The error bars indicate the standard deviation around the mean. The parameter value used in our experiments is marked with a circle. As can be seen our choice of parameters is rather conservative and does not correspond to the best set. This is an indication that the model has not been over-tuned on the given test data.

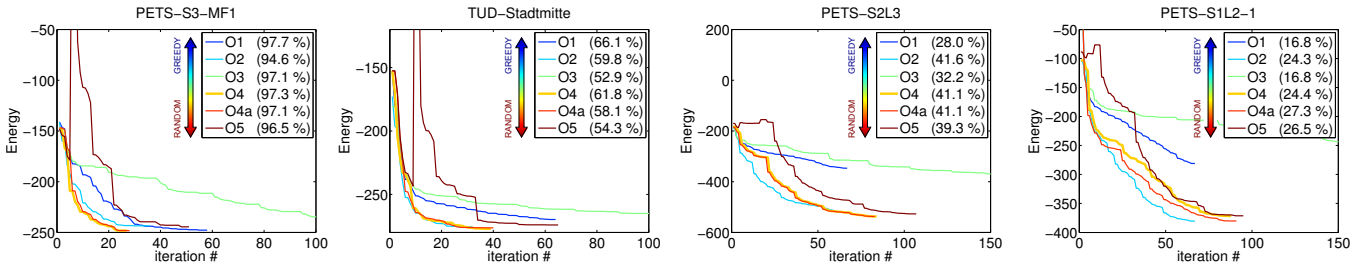

Fig. 10. Energy minimization with different optimization techniques on four exemplar sequences (see text for a detailed explanation of the individual strategies). Our proposed optimization scheme described in section 4 corresponds to *O4*. The final tracking accuracy w.r.t. ground truth is reported in parentheses for each case.

understand this choice, we conduct a set of experiments that vary in the way the jumps are selected and applied. They show that the exact choice is not critical, and that the optimization scheme described in Sec. 4 is a reasonable compromise between fast convergence and low energy. To this end, we compare our results to those obtained with five modified energy minimization algorithms ranging from greedy to random (*cf*. Fig. 10).

To better understand their differences, let us first recall our originally proposed scheme (Sec. 4). We alternate between two distinct algorithms: (1) Purely continuous conjugate gradient descent, which runs until convergence or to a maximal number of iterations (here set to 30, which suffices to get close to a local minimum), and (2) discontinuous jump moves that are executed for all trajectories at once. We now examine the influence of five alternative jump move strategies; the gradient descent is left unchanged. Fig. 10 shows the results.

1) Out of all possible move types and trajectories, the most greedy strategy *O1* always chooses the best possible modification of the current configuration, *i.e.* the one that yields the largest decrease of the energy value. Note that only one trajectory is modified between two continuous optimization runs, which generally leads to slower convergence.
2) The less greedy *O2* chooses the move type that maximally reduces the energy as applied to all trajectories simultaneously, rather than only one as for *O1*. This often leads to a fast energy drop within the first few iterations. However, the reached minimum is usually not as strong as the one found with a more random strategy, such as *O4*.
3) To evaluate the effect of greedily choosing trajectories, *O3* uses a predefined move order. The difference to our method (*O4*) is that instead of modifying all trajectories at once, the best one is picked greedily. This severely limits the possible state space changes. Consequently, the search largely stays within one region of the energy and continuous optimization is not able to descend much further. As a result, this optimization leads to extremely slow convergence.
4) *O4a* also uses a prescribed move order, but modifies all trajectories at each iteration, which significantly speeds up the optimization process. The only difference between our proposed scheme (*O4*) and *O4a* is that a different prescribed order of the jump moves is used. As expected, these two strategies are very close in terms of convergence rate and the achieved results. This shows that the move order does not play a crucial role on average.
5) Finally, *O5* represents the most random strategy. First, the move type is picked randomly each time. Moreover, a 'bad' jump that increases the energy is accepted with probability p, which is in turn decreased with time: $p = e^{-0.05 \cdot \text{iter}}$. This strategy is reminiscent of simulated annealing methods. We find that allowing jumps towards higher energy regions delays the search and does not lead to stronger minima. A more conservative strategy, such as *O4*, finds its way towards regions of a lower energy more quickly and more reliably.

Method	MOTA	MOTP	MT	ML	FM	ID	Rcl	Prc
full	61.4	67.8	11	11	17	24	65.7	95.1
JM only	52.2	62.4	9	11	34	42	62.2	89.3
GD only	41.4	68.2	3	17	14	15	44.7	94.5
EKF	39.8	66.3	3	18	16	13	43.1	94.5

TABLE 2. Average optimization results with disabled gradient descent *(JM only)* vs. disabled jump moves *(GD only)*.

From our results (Fig. 10) we can thus conclude that different optimization schedules lead to minima with a comparable energy. The crucial aspect is to include jump moves to escape weak local minima, since a purely continuous optimization is only able to search within a small local neighborhood of the state space in case of non-convex energies. However, the exact order, frequency, and selection of jumps is of minor importance.

Finally, Tab. 2 shows two further experiments where we either turn off the gradient descent-based optimization and only perform the proposed discontinuous jump moves *(JM only)* or vice versa *(GD only)*. As expected, a purely gradient descent-based optimization only slightly improves the accuracy over the EKF initialization and quickly terminates in a nearby local minimum. On the other hand, the discontinuous jump moves do a good job by sampling varying configurations of the solution space, but are at the same time rather constrained to the present shape of trajectories. Only by combining the two schemes *(full)* is it possible to reach good optima of the proposed energy.

6.3 Datasets

We validate our method on seven challenging, publicly available video sequences. Six of them are part of the PETS 2009/2010 benchmark [17], [18]. We only use the first view of each sequence in all our experiments. They are recorded outdoors from an elevated viewpoint, corresponding to a typical surveillance setup. Note that targets exhibit strong variation in appearance due to shadows and lighting changes. The sequence *S2L1* is the most widely used in multi-target tracking literature. Although it includes non-linear motion of targets, targets in close proximity and a scene occluder, the results seem to saturate on that sequence (tracking accuracy $>90\%$). We therefore extend our test data with two more difficult video sequences with high crowd density (*S2L2* and *S2L3*). To push our tracker to its limits, we additionally use two even more difficult scenarios (*S1L1-2* and *S1L2-1*), which were originally intended for person counting and density estimation, rather than for tracking individuals. In these videos, pedestrians permanently become occluded, providing only little image evidence for our full-body person detector.

Finally, the *TUD-Stadtmitte* sequence [1] is a real-world video filmed in a busy pedestrian street. Here, the size of the pedestrians on the image plane varies significantly. Moreover, the camera is positioned quite low, leading to more complex occlusion patterns and rather inaccurate ground plane locations (due to weak 3D geometry).

The annotated ground truth, the detector responses, and the boundaries of the tracking area for all datasets used here are publicly available on the authors' website.

6.4 Metrics

Conducting an objective comparison of different tracking methods is challenging for various reasons. First, the importance of individual tracking failures is usually application specific and should be weighted accordingly. Second, the definition of a correct or incorrect tracker output may itself be ambiguous, and usually requires an additional parameter (*e.g.*, a threshold) to assess both the correctness and the precision of a tracker.

We follow the currently most widely accepted protocol, the CLEAR MOT [36] metrics, for quantitative evaluation. Since all targets are tracked in 3D space, we compute their distance to the manually annotated ground truth on the ground plane and set the hit/miss threshold to 1m. The *Multi-Object Tracking Accuracy* (MOTA) combines three types of errors – false positives (FP), missed targets (FN), and identity switches (ID) – and is normalized such that the score of 100% corresponds to no errors. All three error types are weighted equally. We also report individual values for all errors, as well as the number of fragmentations (FM) of ground truth trajectories according to [28]. The *Multi-Object Tracking Precision* (MOTP) measures the alignment of the tracker output w.r.t. the ground truth. It reflects the average distance between the output and the ground truth normalized to the hit/miss threshold value. Mostly Lost (ML) and Mostly Tracked (MT) scores are computed on entire trajectories and measure how many ground truth trajectories are lost (tracked for less than 20% of their life span) or tracked successfully (tracked for at least 80%).

6.5 Quantitative evaluation

Table 3 gives the quantitative results for all metrics, computed on all seven sequences individually. We report the results of five methods: The full model including occlusion reasoning and the appearance model, denoted as OM+APP (see also Fig. 11 for a visual illustration). For comparison, we also report results of our method without appearance term, both without (no OM, [4]) and with occlusion modeling (OM, [2]). Note that the results for these two previous methods improve upon those presented in the respective previous publication. The results are compared to those of a state-of-the-art discrete tracker [8], based on the k-shortest Paths (KSP) algorithm on a regular grid as well as to a well-known boosted particle filter (BPF) method [30]. Finally, we report the tracking results of an extended Kalman filter (EKF) (as described in Sec. 4.2) that we use as initialization. Furthermore, we report the average performance across several video sequences. Since the data exhibits a strong variability in person count, we compute the average performance for two separate groups of sequences: An easier set *(G1)*, containing less than 10 individuals per

Sequence	Method	MOTA	MOTP	*GT*	MT	ML	FP	FN	ID	FM	Rcll	Prcsn	Fa/F
PETS-S2L1	OM+APP	90.6	**80.2**	23	**21**	1	59	302	**11**	**6**	92.4	98.4	0.07
(795 frames)	OM	88.6	76.9	23	**21**	**0**	259	**171**	19	12	**95.7**	93.6	0.33
(up to 8 targets)	no OM	**91.6**	79.3	23	**21**	**0**	**53**	262	16	11	93.4	**98.6**	**0.07**
	KSP [8]	80.3	72.0	23	17	2	126	641	13	22	83.8	96.3	0.16
	EKF	68.0	76.5	23	9	1	65	1173	25	30	70.3	97.7	0.08
TUD-Stadtmitte	OM+APP	71.1	65.5	9	**7**	**0**	92	108	4	**3**	84.7	86.7	0.51
(179 frames)	OM	**73.4**	65.0	9	**7**	**0**	83	**102**	3	**3**	**85.6**	87.9	0.46
(up to 5 targets)	no OM	68.0	**67.1**	9	5	1	**49**	172	5	4	75.7	**91.6**	**0.27**
	KSP [8]	45.8	56.7	9	1	1	117	261	5	15	63.1	79.2	0.65
	EKF	58.2	58.3	9	3	**0**	115	172	**2**	6	75.1	81.9	0.65
PETS-S3-MF1	OM+APP	96.7	82.7	7	**7**	**0**	5	**12**	**0**	**0**	**97.7**	99.0	0.05
(107 frames)	OM	94.7	82.6	7	**7**	**0**	12	**12**	3	1	**97.7**	97.7	0.11
(up to 7 targets)	no OM	**97.1**	**83.4**	7	**7**	**0**	3	**12**	**0**	**0**	**97.7**	99.4	0.03
	KSP [8]	83.7	77.8	7	6	1	22	62	**0**	**0**	87.9	95.4	0.21
	EKF	66.7	81.9	7	2	**0**	**0**	169	**0**	1	66.7	**100.0**	**0.00**
PETS-S2L2	OM+APP	56.9	59.4	74	28	12	622	2881	99	73	65.5	89.8	1.43
(436 frames)	OM	**57.2**	59.7	74	**31**	**8**	772	**2684**	120	87	**67.9**	88.0	1.77
(up to 33 targets)	no OM	51.9	60.1	74	18	11	434	3473	115	86	58.4	91.8	1.00
	KSP [8]	24.2	**60.9**	74	7	40	**193**	6117	**22**	**38**	26.8	**92.1**	**0.44**
	EKF	28.6	60.3	74	2	32	280	5565	74	116	32.9	90.7	0.64
PETS-S2L3	OM+APP	**45.4**	64.6	44	9	**18**	169	1572	38	27	51.8	90.9	0.70
(240 frames)	OM	43.9	61.4	44	**11**	20	214	1586	28	22	51.3	88.7	0.89
(up to 42 targets)	no OM	44.1	**65.8**	44	9	22	89	1694	38	22	48.0	94.6	0.37
	KSP [8]	28.8	61.8	44	5	31	45	2269	**7**	**12**	30.4	95.7	0.19
	EKF	20.4	63.3	44	1	35	**13**	2543	8	33	21.1	**98.1**	**0.05**
PETS-S1L1-2	OM+APP	57.9	59.7	36	**19**	11	148	918	21	13	64.5	91.8	0.61
(241 frames)	OM	57.8	61.9	36	18	8	188	875	27	20	66.2	90.1	0.78
(up to 20 targets)	no OM	**59.0**	59.2	36	16	**4**	118	921	22	16	64.4	93.4	0.49
	KSP [8]	51.5	64.8	36	16	14	98	1151	**4**	**8**	55.5	93.6	0.41
	EKF	34.6	63.2	36	3	17	**10**	1664	6	18	35.2	**98.9**	**0.04**
PETS-S1L2-1	OM+APP	30.8	49.0	43	**7**	20	227	2308	61	35	38.5	86.4	1.13
(201 frames)	OM	**31.4**	53.2	43	**7**	**19**	177	2347	51	45	37.4	88.8	0.88
(up to 42 targets)	no OM	26.3	53.5	43	**7**	23	171	2530	64	36	32.6	87.7	0.85
	KSP [8]	19.5	**60.6**	43	4	29	64	2950	**7**	**11**	21.4	**92.6**	0.32
	EKF	9.5	53.1	43	0	34	**38**	3326	28	46	11.3	91.8	**0.19**
mean (G1)	Det (HOG+HOF)	-	-	-	-	-	900.7	158.0	-	-	89.1	60.6	2.7
(low density)	OM+APP	**86.1**	76.1	13.0	**11.7**	0.3	52.0	140.7	**5.0**	**3.0**	91.6	94.7	0.2
	OM	85.6	74.8	13.0	**11.7**	**0.0**	118.0	**95.0**	8.3	5.3	**93.0**	93.1	0.3
	no OM	85.6	**76.6**	13.0	11.0	0.3	**35.0**	148.7	7.0	5.0	88.9	**96.5**	**0.1**
	KSP [8]	69.9	68.8	13.0	8.0	1.3	88.3	321.3	6.0	12.3	78.3	90.3	0.3
	BPF [30]	45.4	68.2	13.0	8.7	0.3	566.7	317.0	34.0	43.7	81.1	70.6	1.5
	EKF	64.3	72.2	13.0	4.7	0.3	60.0	504.7	9.0	12.3	70.7	93.2	0.2
mean (G2)	Det (HOG+HOF)	-	-	-	-	-	1331.8	1919.5	-	-	**56.5**	66.4	4.4
(high density)	OM+APP	**47.8**	58.2	49.2	15.8	15.2	291.5	1919.8	54.8	37.0	55.1	89.7	1.0
	OM	47.6	59.1	49.2	**16.8**	**13.8**	337.8	**1873.0**	56.5	43.5	55.7	88.9	1.1
	no OM	45.3	59.7	49.2	12.5	15.0	203.0	2154.5	59.8	40.0	50.8	91.9	0.7
	KSP [8]	31.0	62.0	49.2	8.0	28.5	100.0	3121.8	**10.0**	**17.2**	33.5	93.5	0.3
	BPF [30]	30.1	**62.7**	49.2	6.2	21.5	257.0	2773.8	91.8	143.5	36.9	88.4	0.8
	EKF	23.3	60.0	49.2	1.5	29.5	**85.2**	3274.5	29.0	53.2	25.1	**94.9**	**0.2**

TABLE 3. Quantitative results on all datasets. Due to the large variability in the number of targets (see annotation), we report averages over the easier (G1, first three datasets) and the four more challenging sequences (G2) separately. We additionally report the average performance of the underlying people detector.

frame, and a more challenging group *(G2)*, where up to 42 pedestrians are present simultaneously.

As expected, explicitly taking occlusion into account increases the overall tracking accuracy *(MOTA)*. However, in less dense tracking scenarios occlusion computation cannot show its benefits, because pedestrians are fully visible most of the time. On the other hand, in crowded environments the accuracy increases by over 2 percentage points on average, and over 5 percentage points in the most difficult case (*PETS-S2L2*). The number of mostly tracked targets rises by 35%, while having almost 10% fewer trajectories that are mostly lost without modeling occlusions.

Compared to our full tracking system including occlusion reasoning (OM), the appearance model forces some parts of the tracks to be removed, thereby raising the amount of missed targets by $\approx 5\%$ on average. At the same time, the number of ID swaps is almost halved for the low density group and still reduced by $\approx 3\%$ in the difficult cases. More prominent is the effect on false alarms. The use of the appearance model weeds out 56% of all false positive detections in less dense scenarios, yielding a false alarm rate of only $.2$ targets per frame. Even though including the appearance model does not lead to higher combined accuracy score in every single case, it turns out to improve the performance on average and must not be ignored when the correct identification of targets is crucial.

For a comparison to tracking on a discrete grid [8], the detections are projected onto the ground plane and the target evidence is distributed to all neighboring cells according to a normal distribution. The corresponding parameters have been manually determined to yield the best possible results. Discrete global optimization clearly outperforms the recursive tracker (EKF) in terms of accuracy, by recovering more trajectories while better keeping track of the target identities. However, the proposed continuous scheme outperforms the discrete tracker on all sequences. Moreover, the spatial discretization limits the achievable precision. This becomes most apparent in the low density setting *(G1)* where targets can be localized more precisely by the detector. Here, the MOTP score is 3.4% lower than that of a Kalman filter and 6.5% lower than our best result (OM+APP).

To compare our method to another baseline we use a recent implementation of the boosted particle filter (BPF) [30] where we tuned the parameters to achieve the best possible results. We only report the average performance on both sets of sequences. While this method recovers substantially more tracks than the Kalman filter, it struggles to suppress persistent false detections which in turn leads to a low precision value.

7 Conclusion

We have presented a continuous energy minimization framework for multi-target tracking, which included explicit occlusion reasoning and appearance modeling. Contrary to many previous non-recursive tracking methods, our aim was to forgo (near) global optimizability and instead model (most of) the crucial aspects of tracking multiple targets as closely as possible. All components are modeled by closed-form, continuously differentiable functions, which allowed for an efficient evaluation of the gradient in closed form. The resulting non-convex energy is minimized by both, a local gradient descent search and a set of discontinuous jump moves. Although the energy can only be minimized locally, an extensive experimental evaluation on several challenging datasets showed that our approach leads to very competitive results, both visually and in terms of quantitative evaluation w.r.t. to ground truth. Although the novel, differentiable appearance model does not lead to a consistent accuracy improvement across all sequences, it significantly reduces the number of false positives and identity switches, which are an important factor in various applications.

In future work we plan to integrate part-based detections into our framework to achieve a higher recall, thereby raising the tracking performance further. Moreover, it is desirable to go beyond hand-crafted energies and turn to machine learning techniques to facilitate and automate the process of finding an appropriate set of parameters, or even the functional form of the energy components from training data. To facilitate that, more extensive annotated data sets need to be created.

References

[1] M. Andriluka, S. Roth, and B. Schiele. Monocular 3D pose estimation and tracking by detection. In *CVPR 2010*.

[2] A. Andriyenko, S. Roth, and K. Schindler. An analytical formulation of global occlusion reasoning for multi-target tracking. In *11th International IEEE Workshop on Visual Surveillance*, 2011.

[3] A. Andriyenko and K. Schindler. Globally optimal multi-target tracking on a hexagonal lattice. In *ECCV 2010*, vol. 1, pp. 466-479.

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

[5] B. Benfold and I. Reid. Stable multi-target tracking in real-time surveillance video. In *CVPR 2011*.

[6] J. Berclaz, F. Fleuret, and P. Fua. Robust people tracking with global trajectory optimization. In *CVPR 2006*.

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

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

[9] J. Black, T. Ellis, and P. Rosin. Multi view image surveillance and tracking. In *Motion&Video Computing Workshop*, Dec. 2002.

[10] 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 *ICCV 2009*.

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

[12] M. Brookes. *The Matrix Reference Manual*, 2005.

[13] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR 2005*, pages 886–893.

[14] M. Eichner and V. Ferrari. We are family: Joint pose estimation of multiple persons. In *ECCV*, volume 6311, pages 228–242, 2010.

[15] M. Enzweiler, A. Eigenstetter, B. Schiele, and D. M. Gavrila. Multi-cue pedestrian classification with partial occlusion handling. In *CVPR 2010*.

[16] A. Ess, B. Leibe, K. Schindler, and L. van Gool. Robust multiperson tracking from a mobile platform. *IEEE T. Pattern Anal. Mach. Intell.*, 31(10):1831–1846, Oct. 2009.

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

[18] J. M. Ferryman and A. Shahrokni. PETS2009: Dataset and challenge. In *Winter-PETS*, Dec. 2009.

[19] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe. Multi-target tracking using joint probabilistic data association. In *IEEE Conf. on Decision and Control*, volume 19, pages 807–812, Dec. 1980.

[20] W. Ge and R. T. Collins. Multi-target data association by tracklets with unsupervised parameter estimation. In *BMVC 2008*.

[21] H. Grabner, J. Matas, L. Van Gool, and P. C. Cattin. Tracking the invisible: Learning where the object might be. In *CVPR 2010*.

[22] P. Green. Reversible jump markov chain monte carlo computation and bayesian model determination.*Biometrika*, 82(4):711–732, 1995.

[23] J.Henriques, R. Caseiro, and J.Batista. Globally optimal solution to multi-object tracking with merged measurements. In *ICCV 2011*.

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

[25] L. Kratz and K. Nishino. Tracking with local spatio-temporal motion patterns in extremely crowded scenes. In *CVPR 2010*.

[26] C.-H. Kuo, C. Huang, and R. Nevatia. Multi-target tracking by on-line learned discriminative appearance models. In *CVPR 2010*.

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

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

[29] S. Oh, S. Russell, and S. Sastry. Markov chain Monte Carlo data association for multi-target tracking. *IEEE Transactions on Automatic Control*, 54(3):481–497, 2009.

[30] K. Okuma, A. Taleghani, O. D. Freitas, J. J. Little, and D. G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *ECCV 2004*, volume 1, pages 28–39.

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

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

[33] M. Rodriguez, I. Laptev, J. Sivic, and J.-Y. Audibert. Density-aware person detection and tracking in crowds. In *ICCV 2011*.

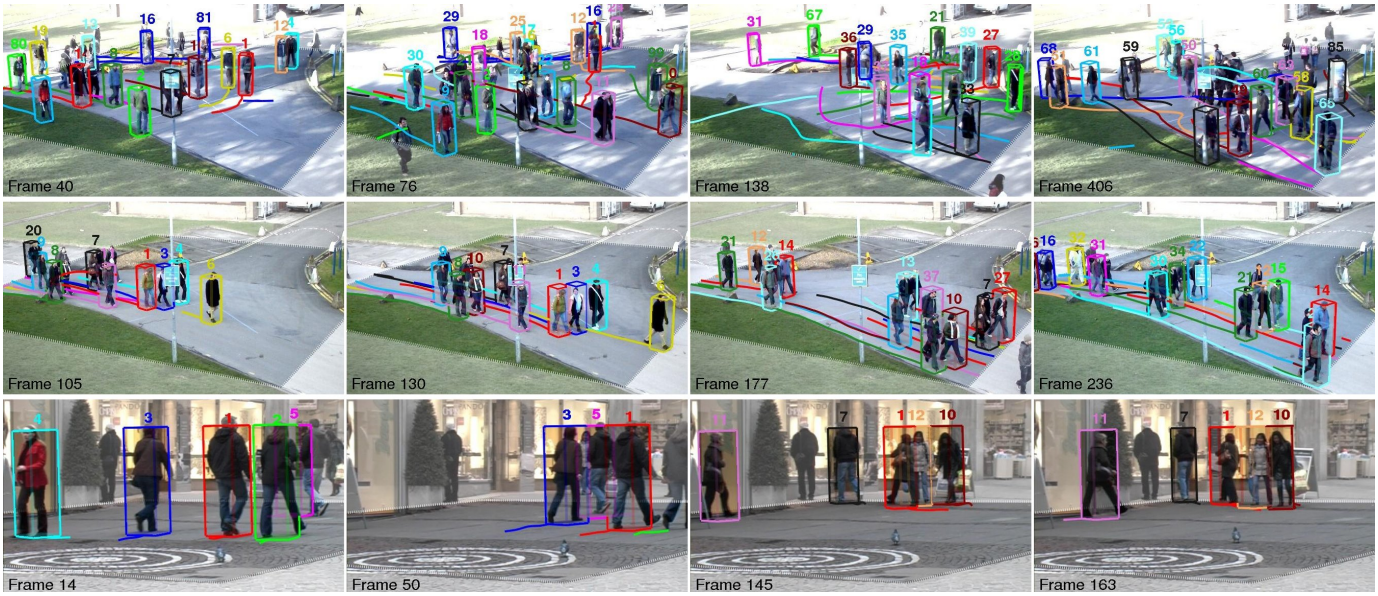

Fig. 11. Tracking results of our method (OM+APP) on (top to bottom) *PETS-S2L2*, *PETS-S1L1-2* and *TUD-Stadtmitte*.

[34] H. B. Shitrit, J. Berclaz, F. Fleuret, and P. Fua. Tracking multiple people under global appearance constraints. In *ICCV 2011*.
[35] L. Sigal and M. J. Black. Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In *CVPR 2006*.
[36] R. Stiefelhagen, K. Bernardin, R. Bowers, J. S. Garofolo, D. Mostefa, and P. Soundararajan. The CLEAR 2006 evaluation. In *CLEAR*, 2006.
[37] J. Vermaak, A. Doucet, and P. Pérez. Maintaining multi-modality through mixture tracking. In *ICCV 2003*.
[38] S. Walk, N. Majer, K. Schindler, and B. Schiele. New features and insights for pedestrian detection. In *CVPR 2010*.
[39] C. Wojek, S. Walk, S. Roth, and B. Schiele. Monocular 3D scene understanding with explicit occlusion reasoning. In *CVPR 2011*.
[40] B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *ICCV 2005*.
[41] Z. Wu, T. H. Kunz, and M. Betke. Efficient track linking methods for track graphs using network-flow and set-cover techniques. In *CVPR 2011*.
[42] J. Xing, H. Ai, and S. Lao. Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses. In *CVPR 2009*.
[43] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *CVPR 2008*.

Anton Milan (né Andriyenko) received his Diplom degree in Computer Science (Dipl.-Inform.) from the University of Bonn in 2008. He has worked as a software developer in the computer graphics industry. In late 2009 he joined the Image Understanding group and in 2010 the Visual Inference group at the Technische Universität Darmstadt where he defended his PhD in May 2013 and currently holds his position as a research assistant. He served as a reviewer for various computer vision conferences and journals. His research interests are energy-based optimization methods for real-world multiple people tracking scenarios.

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 Brown University, and in 2007 the PhD degree in Computer Science from the same institution. Since 2007 he is on the faculty of Computer Science at Technische Universität Darmstadt, Germany (Juniorprofessor 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 an honorable mention for the Marr Prize at ICCV 2005 (with M. Black), 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). In 2013, he was awarded a Starting Grant of the European Research Council (ERC). He served as an area chair for ICCV 2011, ECCV 2012, and CVPR 2013, and is member of the editorial board of the International Journal of Computer Vision (IJCV).

Konrad Schindler received a Diplomingenieur (M.tech) degree in photogrammetry from Vienna University of Technology, Austria in 1999, and a PhD from 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 of Graz University of Technology, the Digital Perception Lab of Monash University, and the Computer Vision Lab of ETH Zurich. He became assistant professor of Image Understanding at TU Darmstadt in 2009, and since 2010 has been a tenured professor of Photogrammetry and Remote Sensing at 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 associate editor for the ISPRS Journal of Photogrammetry and Remote Sensing, and for the Image and Vision Computing Journal.