Multi-target Tracking by Continuous Energy Minimization

Anton Andriyenko
TU Darmstadt
Computer Science Department

Konrad Schindler
ETH Zürich
Photogrammetry and Remote Sensing Group

Abstract

We propose to formulate multi-target tracking as minimization of a continuous energy function. Other than a number of recent approaches we focus on designing an energy function that represents the problem as faithfully as possible, rather than one that is amenable to elegant optimization. We then go on to construct a suitable optimization scheme to find strong local minima of the proposed energy. The scheme extends the conjugate gradient method with periodic trans-dimensional jumps. These moves allow the search to escape weak minima and explore a much larger portion of the variable-dimensional search space, while still always reducing the energy. To demonstrate the validity of this approach we present an extensive quantitative evaluation both on synthetic data and on six different real video sequences. In both cases we achieve a significant performance improvement over an extended Kalman filter baseline as well as an ILP-based state-of-the-art tracker.

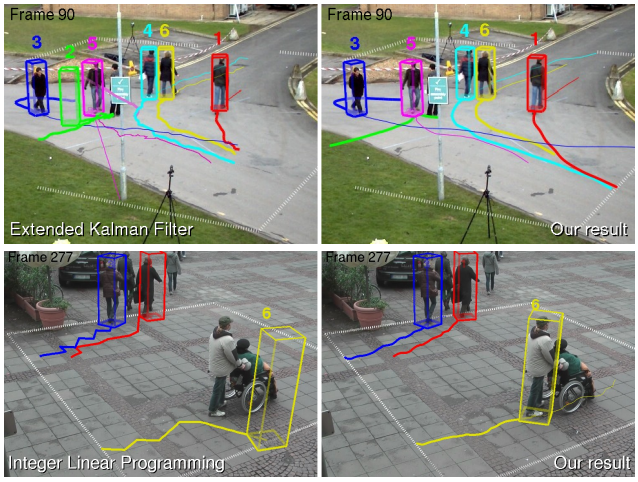

Figure 1. Initial values obtained by an EKF (top left) and an ILP-based tracker (bottom left) and tracking results after global continuous optimization (right). Our method produces smooth, persistent trajectories and significantly reduces the number of false positives and lost targets.

1. Introduction

Multi-target tracking is a now classical, but difficult task in computer vision. Following multiple targets while robustly maintaining data association remains a largely open problem. This is due to several aspects. A main difficulty is the complexity of the state space one has to deal with: the number of possible target trajectories over time is very large (in fact infinite, if the location space is continuous), and there is a trajectory for each of a discrete (but often unknown) number of targets. By itself a huge state space need not be a problem, but several physical constraints introduce dependencies both between different locations of the same target and between different targets. For instance, each object's linear and angular velocity must be physically plausible, and the distance between any two objects cannot become arbitrarily small. Since the discrete trajectories are not independent of each other, maximizing their joint posterior is in general NP-complete. To compound that, inter-object occlusions cause appearance changes and missing evidence.

To resolve between-object interactions, several approaches have recently been proposed which aim to include them in the model and find a joint solution, as opposed to tracking each target individually. This is usually achieved by restricting the state space to a finite set of candidate locations, either by thresholding the observation likelihood or by regularly discretizing the location space. The discretization, together with certain simplifications of the physical constraints, yields energy functions for which a (nearly) global minimum can be found. Although this property is certainly attractive, the price to pay is an energy function which only roughly approximates the underlying posterior.

Here, we raise the question whether it is really advisable to find the global optimum of an inaccurate energy, or whether it may be more appropriate to construct an energy which faithfully represents the actual problem, despite the fact that it is no longer convex (or at least has a convex relaxation). We propose an energy function over all target locations and all frames of a time window, which covers many important aspects of multi-target scenarios. To minimize the resulting energy, we devise a local optimization

scheme which is able to explore many potentially interesting regions of the search space without getting trapped in the initial basin of attraction.

Our approach goes beyond the state-of-the-art in several ways. (1) the targets' locations are not bound to discrete object detections or grid positions, meaning that each target's position is still defined in case of detector failure, and that there is no grid aliasing; (2) there is no need to unnaturally restrict the energy function. Arbitrary object dynamics, appearance models, and even more involved extensions such as group behavior can be integrated into the energy. While we cannot guarantee global optimality, our experiments suggest that the tracking problem does have enough structure that for reasonable energy functions one can avoid weak local minima and find plausible modes of the posterior; (3) the custom-tailored minimization procedure is powerful, yet efficient: it is capable of changing the dimensionality, thereby exploring a much larger portion of the search space than standard gradient methods, but nevertheless stays focused on the promising regions and avoids random search behavior.

The rest of the paper is structured as follows. After discussing related work in Section 2, we present our approach by first defining the global energy function and all its individual components in Section 3.2 followed by an in-depth description of the minimization procedure in Section 3.3. Finally, Section 4 presents quantitative evaluation and experimental results of our approach.

2. Related Work

Object tracking has a long history in computer vision, and a complete review lies beyond the scope of this paper. In this section we concentrate on visual multi-target tracking, with a focus on methods based on optimization.

Compared to tracking a single target, multi-target tracking is a lot more complicated: the single-target case can in most cases be solved by detecting the object in each frame – possibly only within a local region around the predicted position – and "connecting the dots" to a consistent trajectory; for multiple targets the problem is a lot more complex due to the data association problem, and to interactions between different targets (*e.g*. inter-object occlusion). An additional difficulty is that in most scenarios the number of targets is not known a-priori, and may in fact vary over time.

Early work mostly focused on recursive methods, where the current state depends only on the previous one: initially Kalman filtering, *e.g*. [5], and later particle filtering [8, 13, 16], which represents the posterior by a set of samples rather than an analytic expression, and can thus better cope with ambiguous, multi-modal distributions.

Recently, several non-recursive approaches have appeared, which aim to formulate the problem such that a solution can be found which is (in some cases globally) optimal over a longer time interval. One way to reduce the immense solution space of tracking over extended time windows is to commit in advance to a restricted set of possible target locations [7, 10, 11, 20], which are usually found by appearance-based object detection [6, 17] or by background subtraction [14]. The tracker is forced to form trajectories through these locations, without taking into account localization uncertainty. A different approach, which has been pursued in [2, 3, 4], discretizes the space of possible locations to a regular grid, which avoids early commitment to detection results, but instead introduces discretization errors.

The resulting optimization problems are either quadratic integer programs [7, 11], in which case they are solved to local optimality by custom heuristics based on recursive search or graph cuts; or integer linear programs [2, 4, 10], which are solved to near-global optimality through LP-relaxation. An exception is [20], which solves a simplified version of the problem without occlusions to global optimality with a network flow algorithm, then greedily adds occluded targets.

In the present work we investigate the question, whether the restriction to a countably finite state space is really necessary to perform multi-target tracking. It turns out that a well-designed local optimization scheme in a continuous state space can find better solutions, both in terms of visual quality and in terms of standard quantitative measures of tracking accuracy and precision.

3. Model

The aim of our method is to find an optimal solution for multi-target tracking over an entire video sequence. In other words, each target needs to be assigned a unique trajectory for the duration of the video, which matches the target's motion as closely as possible. To this end, we define a global energy function which depends on *all* targets at *all* frames within a temporal window, and thus represents the existence, motion and interaction of all objects of interest in the scene.

Tracking is performed in world coordinates, *i.e*. the image evidence is projected onto the ground plane. Additionally, the evidence is weighted with a height prior to reduce false detections.

3.1. Notation

Before formally defining the energy function we briefly introduce the notation: the state vector $\mathbf{X}$ consists of ground plane coordinates of all targets at all times. The (x, y)-location of target i at frame t is denoted $\mathbf{x}_i^t$. F and N indicate the total number of frames and targets respectively. Note, that in our formulation the position of each target is always defined and considered when computing the energy, even in case of occlusion.

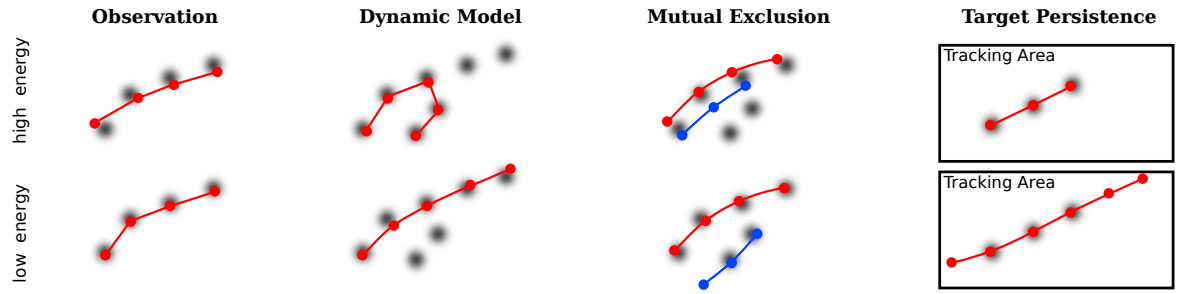

Figure 2. The effects of different components of the energy function. Top row shows a configuration with a higher, bottom row with a lower value for each individual term. Darker gray-values indicate higher target likelihood.

3.2. Energy

There are many possibilities to define an energy (or equivalently, likelihood) function which rewards more plausible configurations and penalizes unreasonable ones. From an optimization perspective it would certainly be beneficial to have a convex function, which by definition only has a single minimum and can be globally optimized independent of initial values. However, the crucial property of an energy function is to approximate the true situation sufficiently well, *i.e*. it should reflect all relevant behaviors which occur in the data as accurately as possible. Otherwise, one is faced with a situation where the minimum may be easily attainable, but no longer corresponds to a useful explanation of the real world. Unfortunately, more realistic formulations of most vision problems lead to highly non-convex energies with many local minima. We argue that in the case of multi-object tracking it is more important to optimize a less contrived, more "correct" energy, and thus waive the requirement that the function be convex. The message of the paper is that nevertheless there is hope – good energy minima can be found.

Our energy function is made up of five terms: an observation term based on image data; three physically motivated priors for object dynamics, collision avoidance and object persistence; and a regularizer (simplicity prior) which tries to keep the number of trajectories low:

$$E(\mathbf{X}) = E_{\text{obs}} + \alpha E_{\text{dyn}} + \beta E_{\text{exc}} + \gamma E_{\text{per}} + \delta E_{\text{reg}}. \quad (1)$$

In the following we describe each component of the energy in more detail. Please refer to Figure 2 for an illustration.

3.2.1 Observation Model

We follow the tracking-by-detection school, *i.e*. the observation at every location is the likelihood of object presence determined by an object detector. Detection has in the last few years proved to be a reliable basis for tracking, and is applicable in unconstrained environments and also with a moving camera. Here, we detect pedestrians with a sliding window approach using both HOG features [6] and histograms of relative optic flow [18]. The energy is smaller if the trajectories pass through regions of high pedestrian likelihood in the individual frames:

$$E_{\text{obs}}(\mathbf{X}) = \sum_{t=1}^{F} \sum_{i=1}^{N} \left(\lambda + \sum_{g=1}^{D(t)} \frac{-c}{\|\mathbf{x}_i^t - \mathbf{d}_g^t\|^2 + c} \right). \quad (2)$$

Here, $D(t)$ is the number of detection score maxima (peaks) in frame t, and $\mathbf{d}_g^t$ is the location of peak g in frame t. The value of λ penalizes existing targets which have no image evidence. It is set to 0.05 for all our experiments.

Approximating the detector output with a sum of Cauchy-like potentials is permissible, since the detection scores by design exhibit smooth fall-off around the peaks.[1] The advantage of the approximation is that one can compute an analytical derivative of E_{obs}, which greatly accelerates minimization.

It is straight forward to extend the observation model beyond detections and also capture targets' appearances. To this end, object colors or histograms in adjacent frames may be compared (favoring smaller variations in color) to better distinguish between individual targets and to avoid identity switches. While in certain situations appearance may act as a strong cue, in our experiments it did not provide enough information to improve the overall performance, probably due to similar clothing and frequent occlusion.

3.2.2 Dynamic Model

For the motion term we use a constant velocity model:

$$E_{\text{dyn}}(\mathbf{X}) = \sum_{t=1}^{F-2} \sum_{i=1}^{N} \left\| \mathbf{v}_i^t - \mathbf{v}_i^{t+1} \right\|^2, \quad (3)$$

where $\mathbf{v}_i^t = \dot{\mathbf{x}}_i^t = \mathbf{x}_i^{t+1} - \mathbf{x}_i^t$ is the current velocity vector of target i. Since maxima of the detector response are in practice not perfectly aligned with targets' locations, the

[1] This is the reason why object detectors have to perform non-maximum suppression on the detector output.

dynamic model can be interpreted as a kind of "intelligent smoothing", which takes into account the other energy terms rather than blindly smooth the nodes of the trajectory curve. It does however go beyond smoothing, for example it helps to prevent identity switches between crossing targets (since it favors straight paths).

Note that the dynamic model has so far been a weak point of trackers based on ILP. These methods suffer from aliasing of the discrete location grid, and either had to discard the dynamic model altogether [4], or had to resort to the weaker constant heading model [2].

3.2.3 Mutual Exclusion

The most obvious physical constraint is that two objects cannot occupy the same space simultaneously. We include this constraint into the energy function by defining a continuous exclusion term:

$$E_{\text{exc}}(\mathbf{X}) = \sum_{t=1}^{F} \sum_{i \neq j} \frac{s_g}{\left\| \mathbf{x}_i^t - \mathbf{x}_j^t \right\|^2} \tag{4}$$

with the scale factor s_g which is set to 35 cm for people tracking. Configurations are penalized where two targets come too close together, and the value goes to infinity when they share one identical position. The term at the same time enforces unique data association (since each detection can only be assigned to one trajectory).

This formulation of collision avoidance takes into account the actual overlap of target volumes and can correctly handle two notoriously difficult problems of multi-target tracking: on the one hand, overlap between targets is checked at *all* times, even if both targets are occluded or otherwise missed by the detector. On the other hand, if two targets would collide due to inaccurate observations, the continuous optimization can push them apart just as much as needed, whereas methods based on grid discretizaton or non-maximum suppression can only "connect the dots" and would have to discard an entire trajectory.

3.2.4 Target Persistence

Another constraint one would in most cases like to integrate into the energy function is the fact that targets cannot appear or disappear within the tracking area (but nevertheless can enter or leave the area). However, we prefer to impose only a soft constraint, since otherwise one would have to explicitly model entry/exit locations (*e.g.* doors) and long term occlusion. Hence the sigmoid penalty

$$E_{\text{per}}(\mathbf{X}) = \sum_{i=1}^{N} \sum_{t \in \{1,F\}} \frac{1}{1 + \exp(1 - q \cdot b(\mathbf{x}_i^t))} \tag{5}$$

is used, where $b(\mathbf{x}_i^1)$ and $b(\mathbf{x}_i^F)$ are the distances of the start, respectively end points of trajectory i to the border of the tracking area. This term enforces merging of existing trajectories through occlusions because an abrupt interruption of a trajectory is penalized.

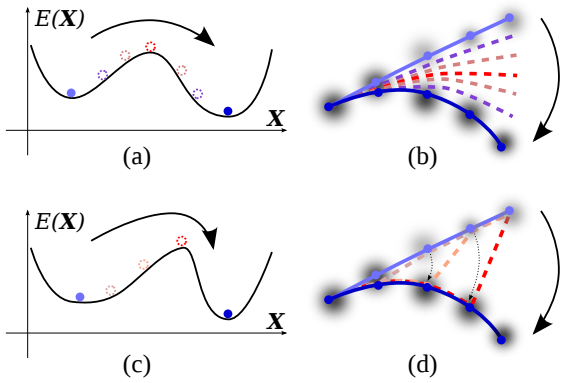

Figure 3. A simple example to illustrate the non-convexity of the continuous tracking formulation. To get from the light blue path (weaker optimum) to the dark blue one (stronger optimum) one has to overcome a ridge of high energy. (a-b) Keeping E_{dyn} low incurs high penalties in E_{obs} as one moves 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 reasonably peaked observation likelihood intermediate cases are even worse.

3.2.5 Regularization

The regularization drives the minimization towards a simpler explanation of the data, *i.e.* a model with fewer targets and longer trajectories:

$$E_{\text{reg}}(\mathbf{X}) = N + \sum_{i=1}^{N} \frac{1}{F(i)}, \tag{6}$$

where $F(i)$ is the temporal length of trajectory i in frames. The regularization balances the model's complexity against its fitting error, and discourages overfitting, fragmentation of trajectories, and spurious identity changes.

3.3. Energy Minimization

The proposed energy is clearly not convex. In fact, a realistic energy, which describes the true situation well, is unlikely to be convex: it is easy to construct examples, which have two equally likely minima separated by a ridge of high energy, *cf*. Fig. 3 for an illustration. The reason for this behavior is the high-order dependence between variables caused by physical constraints.

To mitigate the problem we introduce a number of jump moves which change the dimension of the current state $\mathbf{X}_{curr}$, thereby jumping to a different region of the search space, while still decreasing the energy, *cf*. Fig. 4. In the example it is possible to remove a weak trajectory entirely and initialize another one, while always lowering the energy.

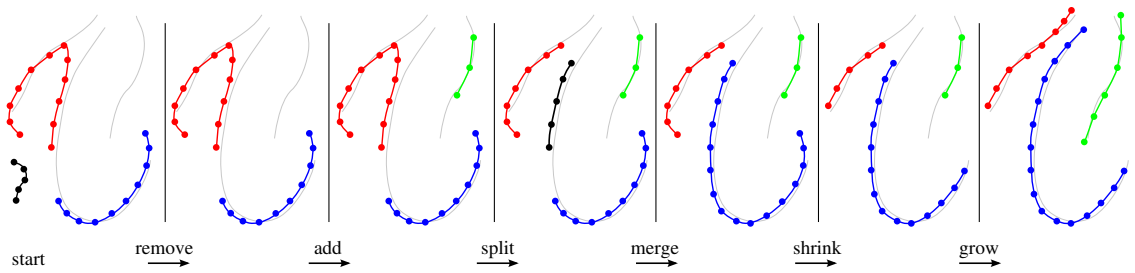

Figure 4. The proposed jump moves give the continuous optimization a higher degree of flexibility allowing a variable number of targets. Even a poor initial configuration can be used to recover the true trajectories. The ground truth is rendered in gray.

To minimize the energy function (1) locally, we use the standard conjugate gradient method. Every n iterations a jump move is executed (unless it would increase the energy). Based on our experience, the order of the jump moves does not influence the final result because the optimization is always able to perform an inverse move to find the way towards a lower energy. The jumps give the optimization a high degree of flexibility – the initial solution need not even have the correct number of targets.

The data-driven strategy to change the dimension of the state vector is reminiscent of reversible jump Markov Chain Monte Carlo methods [9]. However, in contrast to Monte Carlo sampling, our method is deterministic: it exploits the advantages of gradient descent over sampling within one mode, and performs jumps according to a prescribed schedule and only if they decrease the energy.

Growing and Shrinking. Each trajectory can be extended in space-time using standard extrapolation. In contrast, paths can be shortened if not enough image evidence is available. These two steps help to pick up targets missed due to tracker failure and weed out false positives.

Splitting and Merging. To eliminate identity switches, trajectories can be split and merged. Splitting is implemented by breaking paths into two if the split yields lower energy. Merging is executed if two paths can be connected into one with lower energy, preserving physically plausible target motion. Especially the latter is a powerful move 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, but can grow or merge with existing ones at a later iteration. An entire trajectory is removed from the scene if its total energy is positive (meaning that its presence reduces the overall likelihood of the current state, rather than increasing it). Adding again helps to clean up detector failure, whereas removal discards trajectories which have been pushed to a state with little evidence, unreasonable dynamics, and/or overlap with other trajectories.

Initialization. Like any non-convex optimization, the result depends on the initial value from which the iteration is started. However, the described "intelligent 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, which lowers the energy.

Empirically, even the trivial initialization with no targets at all works reasonably well, however it takes many iterations to converge. Instead, we propose to rather use the output of an arbitrary simpler tracker as a more qualified initial value. In our experiments we have used both per-target Extended Kalman filters (EKFs), and a globally optimal discrete tracker based on integer linear programming [2]. In both cases, the trackers are run with different parameters to generate a set of starting values. For both initializations the proposed minimization scheme consistently manages to substantially reduce the energy, and in our experiments has always improved tracking accuracy, *cf*. Tables 1 and 2. Usually different starting values converge to similar (albeit not identical) solutions, see Fig. 5.

Figure 5 shows the convergence behavior for several optimization runs on the same dataset but with varying starting points. Note that a decrease in energy corresponds very well to better tracking performance – an indication that the energy function indeed is a good representation of the true objective. Figure 1 shows a qualitative comparison between initialization (left) and the result after continuous global optimization (right) for two initial values obtained with different techniques. The proposed energy minimization scheme is able to successfully recover persistent trajectories while not suffering from spatial discretization.

4. Experiments

In section 3 we have proposed an energy function which has been conceived with the primary goal to accurately to

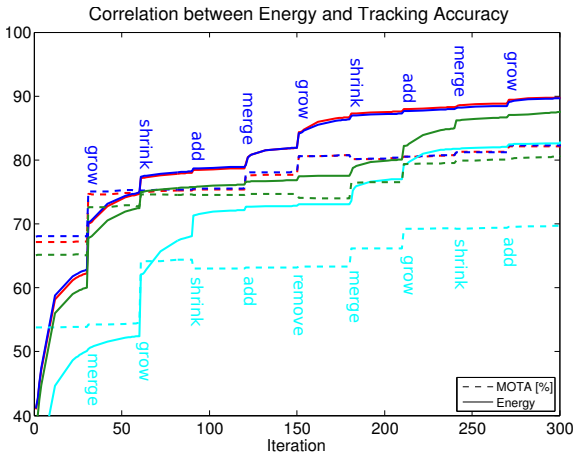

Figure 5. The proposed energy (solid) correlates well with tracking performance w.r.t. ground truth (dashed). Each color represents a different initialization. Energy values have been negated and scaled to better fit the figure.

reflect the actual behavior of multiple interacting targets, *cf*. Fig. 5. 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 have performed an extensive experimental evaluation on various different datasets.

Synthetic dataset. Since it is notoriously difficult to get accurate ground truth for tracking, we first validate our method on synthetic data, for which perfect ground truth is available to quantitatively evaluate the tracking. To emulate realistic trajectories, the dataset is constructed by combining 645 short snippets randomly sampled from real annotated data. To simulate detector failure, 15% of all detections were removed. In our simulations, the minimization in *all* cases lead to a significant improvement in tracking accuracy compared to the initialization. In many cases the ideal result was found, yielding 100% accuracy. In particular, continuous optimization also significantly improved the output of a state-of-the-art discrete tracker based on ILP [2]. Both the ILP-tracker and the subsequent continuous optimization were fed the identical detector evidence, in order to guarantee that the comparison is not biased by the implementation of the detector. In all experiments with synthetic data the *multi-object tracking precision* (see below) was above 95%. This shows a direct benefit of the continuous solution as opposed to a discrete one, which hardly ever exceeds 70% as a consequence of discretization errors.

Real data benchmarks. We have done experiments on four widely used real world datasets as well as on two of our own sequences. All data is recorded outdoors in an unconstrained environment and exhibits strong variability in video quality, image resolution and frame rate.

For multi-view tracking we use the sequences *terrace1* and *terrace2* [3], each containing 2000 frames recorded from four different viewpoints.[2] The videos show up to 6 people walking freely around a small area and feature a lot of occlusions and significant scale changes. Sequence *S2L1* is taken from the VS-PETS 2009 benchmark.[3] Only the first viewpoint is used. The video is filmed with ≈ 7 fps from an elevated viewpoint and is 795 frames long, and shows up to 8 people. The *TUD-Stadtmitte* dataset [1] contains only 179 frames but is very challenging due to the extremely low camera angle – which makes 3D position estimation very difficult – and heavy inter-object occlusion.[4] Finally, we present experiments on a new dataset, which we plan to release in near future. The *ped1* sequence, *cf*. Fig. 1 bottom, is 1400 frames long and shows pedestrians, bicycles and a wheel chair from two viewpoints in a busy pedestrian street. In our experiments we treat each viewpoint separately.

We are able to run the optimization with ≈ 1 sec/frame on entire sequences in all datasets without the need to resort to sliding temporal windows.

4.1. Quantitative Evaluation

There is no single established protocol how to measure multi-object tracking performance. We follow the current best practice and calculate the CLEAR-metrics introduced by [15]. All figures are computed in 3D with a hit/miss threshold of 1 meter. The Multiple Object Tracking Accuracy *(MOTA)* takes into account false positives, missed targets and identity switches. The Multiple Object Tracking Precision *(MOTP)* is simply the average distance between true and estimated targets. Furthermore, we also compute the metrics proposed in [12], which counts the number of mostly tracked *(MT)*, partially tracked *(PT)* and mostly lost *(ML)* trajectories as well as the number of track fragmentations *(FM)* and identity switches *(IDS)*.

4.2. Example Results

Tables 1 and 2 present quantitative results of our approach on all datasets. As initializations we use the solutions delivered by a recent implementation of the classical EKF [19], respectively by the already mentioned ILP tracker. The numbers given are the performance metrics for the initial values (which are by themselves state-of-the-art multi-target trackers and thus serve as baseline), and for the final values found with our proposed formulation. Additionally, we also show the differences between initial and final results. The average is taken over nine different starting

[2] http://cvlab.epfl.ch/data/pom
[3] http://www.cvg.rdg.ac.uk/PETS2009
[4] http://www.mis.tu-darmstadt.de/node/428

points, generated through slight variations of the respective algorithm's parameters. As expected, our proposed method consistently reduces tracking errors and thus improves the average tracking performance in all cases. The slight increase of track fragmentations and ID-switches can be explained by the larger number of successfully tracked targets.

Table 3 presents quantitative results for tracking configurations with the lowest energy for each dataset. Figure 6 shows some example results in three different sequences. Targets moving within a specified tracking area (marked with a dotted line) are successfully tracked over time, with new targets initialized automatically.

The weighting parameters α through δ have been determined empirically and are set to (0.05, 1, 0.5, 0.25) in all our experiments.

Sequence	MOTA	MOTP	MT	PT	ML	FM	IDS
terrace1	87.2 %	79.3 %	6	2	1	10	17
terrace2	88.1 %	78.1 %	7	1	1	11	11
TUD	60.5 %	65.8 %	6	3	0	4	7
PETS	81.4 %	76.1 %	19	4	0	21	15
ped1-c1	63.1 %	80.3 %	15	7	1	4	4
ped1-c2	48.0 %	75.7 %	10	10	2	19	4

Table 3. Quantitative results of our method. For each sequence, the optimization was initialized with multiple EKF and/or ILP trackers with different parameters. The displayed results correspond to the optimum with the lowest energy.

5. Conclusion and Future Work

We have presented an algorithm for jointly tracking a varying number of targets with continuous optimization. Contrary to a recent trend we have shown that convexity[5] is, in the case of tracking, not necessarily the primary requirement for a good cost function: to achieve meaningful (although local) energy minima it is not necessary to limit the state space, as most multi-target trackers implicitly do – either by per-frame non-maxima suppression or by discretizing locations to a coarse grid. Through the minimization of a continuous global energy function, using gradient descent together with appropriate trans-dimensional jump moves, we improve over state-of-the-art multi-target tracking techniques on many public datasets.

In future work we plan to incorporate more sophisticated appearance and dynamic models, and to handle visibility more explicitly. Although our method tracks remarkably well even through occlusions, we believe that explicit occlusion reasoning will help to handle more difficult target interactions with missing detections, crowded scenes and long-term occlusions. Furthermore we hope to be able to reach real-time performance with an even faster optimization based on multi-grid search, as well as a more efficient implementation. This would make the method applicable to real-time applications, by repeatedly solving for only the past frames.

[5] Respectively submodularity for discrete functions.

References

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

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

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

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

[5] J. Black, T. Ellis, and P. Rosin. Multiview image surveillance and tracking. In *Motion&Video Computing Workshop*, 2002.

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

[7] A. Ess, B. Leibe, K. Schindler, and L. Van Gool. A mobile vision system for robust multi-person tracking. In *CVPR'08*.

[8] J. Giebel, D. Gavrila, and C. Schnörr. A Bayesian framework for multi-cue 3d object tracking. In *ECCV*, 2004.

[9] P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

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

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

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

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

[14] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, 1999.

[15] R. Stiefelhagen, K. Bernardin, R. Bowers, J. S. Garofolo, D. Mostefa, and P. Soundararajan. The clear 2006 evaluation. In R. Stiefelhagen and J. S. Garofolo, editors, *CLEAR*, volume 4122 of *LNCS*, pages 1–44. Springer, 2006.

[16] J. Vermaak, A. Doucet, and P. Perez. Maintaining multi-modality through mixture tracking. In *ICCV*, 2003.

[17] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. *IJCV*, 2005.

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

[19] C. Wojek, S. Roth, K. Schindler, and B. Schiele. Monocular 3D scene modeling and inference: understanding multi-object traffic scenes. In *ECCV*, 2010.

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

Sequence	MOTA [%]			MOTP [%]			MT			PT			ML			FM			IDS		
	initial	final	**diff**	initial	final	**diff**	init	fin	**diff**	init	fin	**diff**	init	fin	**diff**	initial	final	**diff**	initial	final	**diff**
TUD	53.3	60.9	**+7.6**	57.4	65.9	**+8.4**	5	6	**+1**	4	3	**-1**	0	0	**+0**	3.1	4.4	**+1.3**	3.8	6.0	**+2.2**
PETS	64.7	78.7	**+14.0**	75.4	76.7	**+1.4**	9	16	**+7**	14	6	**-8**	0	0	**-0**	25.2	19.2	**-6.0**	17.7	14.2	**-3.4**
ped1-c1	32.3	49.7	**+17.4**	78.5	78.1	**-0.4**	1	13	**+12**	14	5	**-9**	8	4	**-3**	1.2	3.6	**+2.3**	1.2	2.7	**+1.4**
ped1-c2	29.0	37.7	**+8.7**	71.8	77.1	**+5.3**	0	7	**+7**	17	12	**-6**	5	4	**-1**	9.2	16.3	**+7.1**	5.9	2.9	**-3.0**
mean	44.8	56.7	**+11.9**	70.8	74.5	**+3.7**	4	11	**+7**	12	6	**-6**	3	2	**-1**	9.7	10.9	**+1.2**	7.1	6.4	**-0.7**

Table 1. Quantitative results of our method. For each metric we report three values: the starting point obtained with an EKF-tracker, our result after energy minimization and their difference (the averages for MT, PT and ML have been rounded for better readability). The tracking accuracy, respectively the number of mostly tracked targets, consistently improve for all datasets. Tracking precision remains the same or improves slightly. Our EKF implementation only supports monocular video, hence, the datasets *terrace1* and *terrace2* could not be tested.

Sequence	MOTA [%]			MOTP [%]			MT			PT			ML			FM			IDS		
	initial	final	**diff**	initial	final	**diff**	init	fin	**diff**	init	fin	**diff**	init	fin	**diff**	initial	final	**diff**	initial	final	**diff**
terrace1	82.8	84.9	**+2.1**	74.3	79.6	**+5.3**	7	7	**-0**	1	1	**+0**	1	1	**-0**	10.4	10.4	**+0.0**	14.4	19.6	**+5.1**
terrace2	75.1	83.8	**+8.7**	71.7	76.7	**+5.1**	9	7	**-2**	0	1	**+1**	0	1	**+1**	7.8	10.6	**+2.8**	11.7	17.1	**+5.4**
TUD	10.2	41.9	**+31.8**	45.3	61.7	**+16.5**	0	3	**+2**	2	6	**+5**	7	0	**-7**	1.0	5.8	**+4.8**	0.8	6.0	**+5.2**
PETS	26.0	42.8	**+16.7**	67.3	74.7	**+7.4**	1	6	**+5**	14	11	**-3**	8	6	**-2**	14.3	17.6	**+3.2**	17.2	14.0	**-3.2**
ped1-c1	43.8	58.9	**+15.1**	72.3	79.9	**+7.6**	9	14	**+5**	6	6	**-0**	8	3	**-5**	3.0	3.5	**+0.5**	3.8	2.5	**-1.2**
ped1-c2	29.1	33.1	**+4.0**	64.2	72.1	**+7.9**	0	7	**+7**	19	14	**-5**	3	2	**-2**	9.4	10.5	**+1.1**	6.6	7.8	**+1.1**
mean	44.5	57.6	**+13.1**	65.8	74.1	**+8.3**	4	7	**+3**	7	7	**0**	5	2	**-3**	7.7	9.7	**+2.0**	9.1	11.2	**+2.1**

Table 2. Quantitative results of our method using the discrete ILP-tracker [2] as initialization. Continuous optimization consistently improves tracking accuracy and tracking precision for all tested sequences.

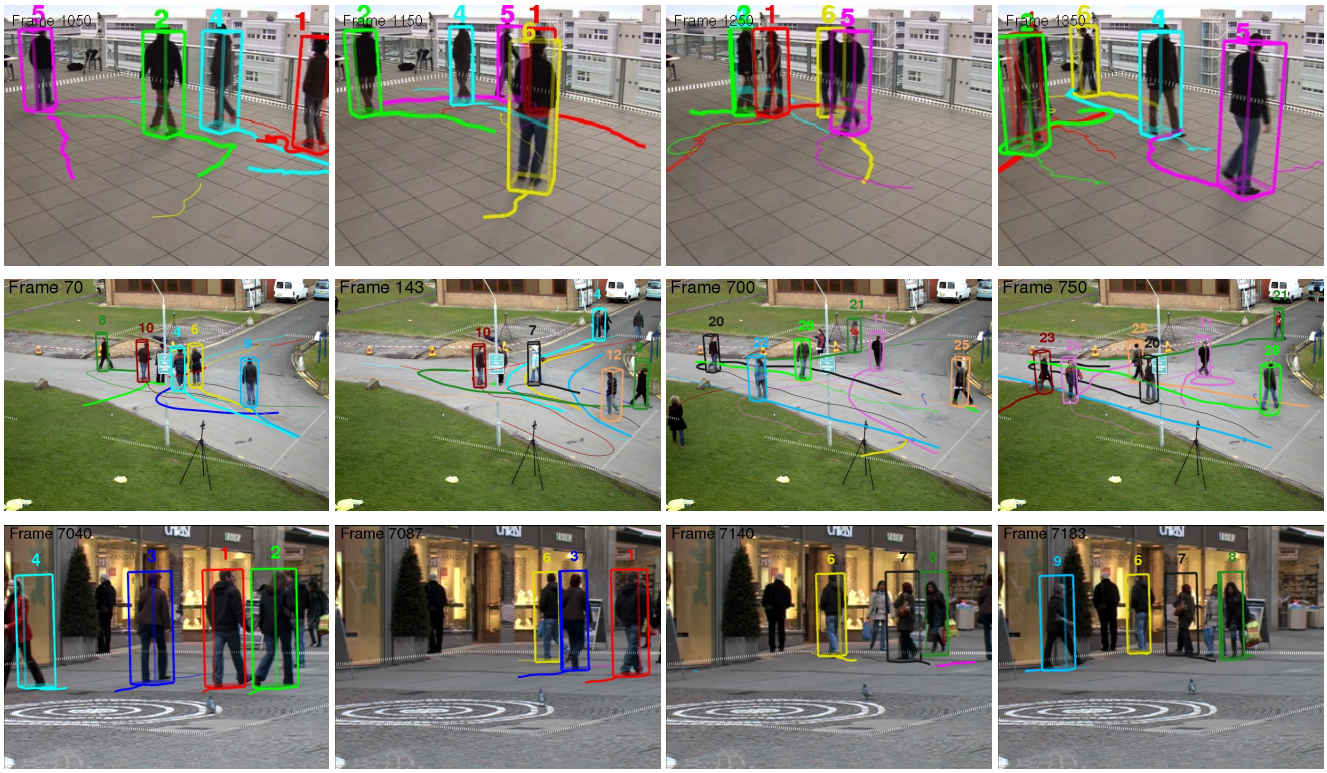

Figure 6. Tracking results obtained with our algorithm. Top to bottom: datasets *terrace1*, *PETS-L2S1* and *TUD-Stadtmitte* with marked tracking area (dotted). Four sample frames are displayed for each dataset from left to right. Trajectories are visualized in bold (past) and thin (future) lines. The identities of tracked targets are color coded.