Globally Optimal Multi-target Tracking on a Hexagonal Lattice

Anton Andriyenko[1] and Konrad Schindler[1,2]

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

Abstract. We propose a global optimisation approach to multi-target tracking. The method extends recent work which casts tracking as an integer linear program, by discretising the space of target locations. Our main contribution is to show how dynamic models can be integrated in such an approach. The dynamic model, which encodes prior expectations about object motion, has been an important component of tracking systems for a long time, but has recently been dropped to achieve globally optimisable objective functions. We re-introduce it by formulating the optimisation problem such that deviations from the prior can be measured independently for each variable. Furthermore, we propose to sample the location space on a hexagonal lattice to achieve smoother, more accurate trajectories in spite of the discrete setting. Finally, we argue that non-maxima suppression in the measured evidence should be performed during tracking, when the temporal context and the motion prior are available, rather than as a preprocessing step on a per-frame basis. Experiments on five different recent benchmark sequences demonstrate the validity of our approach.

1 Introduction

Multi-target tracking in video sequences is a fundamental task of computer vision and video processing, with applications in surveillance, semantic video search, driver assistance, and many more. From a high-level point of view, the aim is to estimate the spatial trajectories of a number of targets over time, i.e. the task is solved when the locations of all targets at each time step are known.

Compared to single-target tracking, the multi-target problem poses additional difficulties: *data association* needs to be solved, i.e. it has to be decided which observation corresponds to which target; and *constraints* between targets need to be taken into account – most importantly, no two targets can occupy the same space at the same time. In probabilistic terms, one aims to maximise the joint posterior of several variables, which are not independent. That posterior depends on two factors: an *observation model*, which measures the agreement between the observed image evidence and the expected appearance of a target; and a *dynamic model*, which measures the agreement between a trajectory and the expected motion pattern of a target.

In the recent past, a main research challenge in multi-target tracking has been to develop schemes which are able to find (nearly) *global* maxima of the

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part I, LNCS 6311, pp. 466–479, 2010.

posterior over the set of trajectories. Their common characteristic is that, in order to enable global optimisation, the set of permissible target locations has to be restricted to a manageable finite set. A-priori the set of possible locations is infinite, or at least very large. There are two main strategies to restrict it to a reasonably small set: either candidate locations are found by thresholding and/or non-maxima suppression of the observation likelihood; or the tracking region is sampled on a regular grid.

The dominant strategy so far has been the first one: the image evidence – typically the output of object detection or background subtraction – is used to identify the most promising target locations per frame. These serve as input for the tracker, which links them to trajectories. A limitation of this strategy is that candidate locations are implicitly assumed to correspond perfectly with true target positions; there is no concept of localisation uncertainty. Another problem is that the space is sampled *only* at promising locations, hence target locations are not even defined in case of missing evidence (e.g. if two targets were both missed by the observation model, it is no longer checked whether they would collide in that frame).

The regular discretisation is attractive, because it is more generic, and because it avoids intermediate hard decisions based on partial evidence, thus allowing for principled probabilistic modelling. A disadvantage is that to keep tracking computationally tractable the grid needs to be significantly coarser than typical image resolutions, and therefore introduces aliasing. A particularly undesirable consequence of the discretisation is that the space is no longer isotropic – the smoothness of a trajectory depends on its alignment with the grid, and jagged trajectories complicate the usage of reasonable dynamic models, which favour smooth motion.

In this paper, we present a global optimisation approach to multi-target tracking on a regular grid, with an a-priori *unknown* number of targets. Original contributions of the work are

- We "re-introduce" the dynamic model, which has traditionally been an integral part of tracking, but in previous work had to be dropped to achieve objective functions, which can be solved to (near) global optimality. Specifically, we include the constant heading prior.
- To best utilise the dynamic model and achieve smoother, more accurate trajectories despite the discrete setting, we propose to use a hexagonal sampling of the location space, rather than a rectangular one.
- We perform non-maxima suppression during tracking rather than independently in every frame, allowing the tracker to recover the most likely locations in the light of *all* evidence, rather than the locally best guess per frame.

Despite the proposed extensions the resulting maximisation of the posterior can still be written as an integer linear program (ILP), by an extension to the formulation of [1]. The ILP is solved efficiently through a linear programming relaxation, in most cases to global optimality.

2 Related Work

Multi-target tracking algorithms can be roughly classified as either recursive methods which base their estimate only on the state of the previous frame, or methods which seek optimality over an extended period of time. Recursive methods rely on a first-order Markov assumption, usually using either Kalman filtering, e.g. [2,3], or – in the presence of more complex posteriors – particle filtering, e.g. [4,5,6]. A different strategy is to aim for an optimal solution over multiple frames. To this end the state space is restricted to a discrete number of possible target locations, either by heuristics based on the single-frame target likelihood, e.g. [7,8,9,10], or by sampling locations on a regular grid, e.g. [11,1].

The more popular strategy has so far been to use single-frame heuristics. After measuring the likelihood that a target is present at any given image location – in most cases by variants of object detection [12,13] or background subtraction [14] – the likelihood function is thresholded and/or its local maxima are found; possible object locations are restricted to these maxima. The optimisation then chooses the best set of trajectories over time, based on the selected locations. Depending on the formulation, this leads to an ILP which is solved by relaxation [8], an integer quadratic program which is solved with problem-specific search procedures [7], or a network flow problem [10], which is in fact closely related to ILP, c.f. [15]. The pruning strategy would be entirely sufficient if the per-frame processing were entirely correct. In practice it has an important shortcoming: the evidence will never be perfect, so that the discrete set after pruning will suffer from false positives (spurious maxima), false negatives (missing maxima), and localisation errors (displaced maxima).

To still restrict the state space, without relying on the per-frame measurements, it has therefore been proposed to sample locations on a regular grid rather than at the modes. Research into grid-based trajectory optimisation started with methods which greedily aim for an optimal trajectory *per target*, e.g. using dynamic programming [11]. In the radar tracking literature, it has been shown long ago how to extend the dynamic programming approach to simultaneously track multiple targets [16], however in practice the computational complexity is prohibitive. An important step forward, which has also inspired our work, is the recent work of Berclaz et al. [1]. Tracking is performed in a globally optimal manner on a regular grid, by casting the problem as an ILP, again solved through relaxation. Contrary to [8] the number of targets need not be known in advance, which is achieved by adding source and sink nodes that can spawn, respectively terminate, trajectories, similar to [10].

This elegant formulation has two main limitations: firstly, in order to arrive at the ILP, the dynamic model had to be discarded. Object dynamics, which are an important component of tracking, are included only in a simplistic way, by allowing arbitrary motion within a grid point's 8-neighbourhood. Secondly, we found that a very peaky observation likelihood with sharp maxima at single grid locations is required for the method to work well. As soon as the object evidence

is blurry and "connecting the dots" becomes ambiguous, the method tends to instantiate multiple trajectories for the same target. This is a price the method pays for the desirable property that it allows for a variable number of targets.

The goal of the present work is to remedy these shortcomings by extending the model appropriately, while still keeping it linear, and hence amenable to global optimisation.

3 Model

In the following we give a detailed description of the proposed multi-view tracking method. We start with the formulation of maximum a-posteriori trajectory estimation as an integer linear program, which is an extension of the formalism introduced by [1]. Next, we introduce our observation model, a probabilistic variant of *tracking-by-detection* designed for tracking targets observed from multiple viewpoints in world coordinates. Furthermore we propose to include non-maxima suppression in the tracker, rather than viewing it as a preprocessing step. We then write the dynamic model as a local soft constraint, by penalising the changes between consecutive motion vectors. In this form it can be re-introduced into the ILP-formulation of multi-target tracking. Finally we move to an important technical issue: in the discrete setting the dynamic model suffers from grid aliasing, hence it is a lot more effective to quantise locations to a hexagonal rather than a rectilinear grid.

3.1 Tracking as Integer Linear Program

To set the stage, we extend the ILP-formulation of multi-target tracking introduced in recent work [8,10,1] for our purposes. The possible target locations are discretised to a finite set of sites $\mathbf{x}_i = (x_i, y_i)$. Among those sites a neighbourhood system $\mathcal{S}$ is defined, where a site's neighbours $\{\mathbf{x}_j : j \in \mathcal{S}(i)\}$ are all sites that can be reached from $\mathbf{x}_i$ in a single time step, including $\mathbf{x}_i$ itself.

Next, we define a *tracklet* X^t_{ijk} as an allowable path over 3 consecutive frames, i.e. a set of three sites $X^t_{ijk} = \{\mathbf{x}^{t-1}_i, \mathbf{x}^t_j, \mathbf{x}^{t+1}_k\}$ such that $j \in \mathcal{S}(i)$ and $k \in \mathcal{S}(j)$. The set of all index triplets (ijk) that produce a valid tracklet is denoted $\mathcal{T}$. The tracklets are the variables of our optimisation problem. They take on values $X^t_{ijk} \in \{0, 1\}$, where $X^t_{ijk} = 1$ means that tracklet (ijk) is part of some trajectory, and $X^t_{ijk} = 0$ means that it is not part of any. The reason for introducing the tracklets is that the dynamic model cannot be included efficiently when operating directly on the sites $\mathbf{x}^t_i$, as will become clear in Sec. 3.4.

Based on the observed evidence $\mathbf{R}$, each tracklet is assigned a goodness-of-fit $u^t_{ijk} = \log \frac{P(X^t_{ijk}=1|\mathbf{R})}{P(X^t_{ijk}=0|\mathbf{R})}$ which compares the hypotheses $X^t_{ijk} = 1$ and $X^t_{ijk} = 0$ in the light of the observation model (Sec. 3.2) and the dynamic model (Sec. 3.4). Thus, multi-target tracking becomes maximising the posterior by picking the best set of tracklets $\mathbf{X}^*$ from $\mathcal{T}$, under two constraints:

1. *collision avoidance*: no two tracklets can have the same midpoint $\mathbf{x}^t_j$; whenever a tracklet X^t_{ijk} is selected, all other tracklets X^t_{qjr} must be discarded.

2. *continuity*: tracklets must form continuous trajectories – whenever a certain tracklet is used in a solution, $X^t_{ijk} = 1$, there must be exactly one tracklet X^{t+1}_{jkl} in the next time step, which is also used. Targets entering or leaving the tracking area are modelled by two virtual *source* and *sink* sites, which are neighbours of all boundary sites and can emit, respectively absorb, targets.

The MAP estimation amounts to the following optimisation problem with the vector $\mathbf{X}$ of all tracklets X^t_{ijk} as argument:

$$U^* = \max_{\mathbf{X}} \sum_{ijk \in \mathcal{T}, t} \left(u^t_{ijk} \cdot X^t_{ijk} \right) \tag{1}$$

$$\text{s. t. } \forall\, ijk \in \mathcal{T}, t: \quad \sum_{q:qjk \in \mathcal{T}} X^t_{qjk} = \sum_{r:jkr \in \mathcal{T}} X^{t+1}_{jkr} \qquad \text{(continuity)} \tag{2}$$

$$\sum_{q,r:qjr \in \mathcal{T}} X^t_{qjr} \leq 1 \qquad \text{(collision avoidance)} \tag{3}$$

$$X^t_{ijk} \in \{0, 1\} \qquad \text{(domain of variables)} \tag{4}$$

Optimisation. Maximising Eq. (1-4) is an integer linear program, and hence NP-complete. However, it can be relaxed to a linear program by replacing the condition $X^t_{ijk} \in \{0, 1\}$ with $0 \leq X^t_{ijk} \leq 1$. The relaxed problem can be efficiently solved with the simplex algorithm or an interior-point method. Moreover, if all variables X^t_{ijk} at the relaxed optimum $\mathbf{X}^*_{LP}$ take on integer values, then it is also a global optimum of the original problem, $\mathbf{X}^*_{LP} = \mathbf{X}^*_{ILP}$. In practice, this happens in most cases. Even if the solution is not completely integral, then in practice the optimality gap is small, and only a tiny fraction of non-integer variables remains (in our experiments $< 0.2\%$), and these are clustered in relatively small connected components of the neighbourhood system. Hence, an optimum of the ILP can be found using the branch-and-cut method with the relaxation as bounding function ("mixed integer programming"), or by "probing", i.e. rounding some non-integer values and solving for the others while monitoring the objective value U (a similar strategy is known as QPBO-P in the graph-cuts context [17]).

To gain some intuition why the LP-solution $\mathbf{X}^*_{LP}$ is largely integral, and amenable to probing or bounding, it is instructive to look at the behaviour of simple paths connecting the source to the sink (see also Fig. 1):

- trivially, all tracklets on a junction-free path Q have the same value X_Q because of the continuity constraint; X_Q will always be integral, because the total contribution of the path to the objective value is $X_\mathsf{Q} \sum u_\mathsf{Q}$, which attains its maximum at $X_\mathsf{Q} = 0$ for $\sum u_\mathsf{Q} \leq 0$, and at $X_\mathsf{Q} = 1$ for $\sum u_\mathsf{Q} > 0$.
- if a path were to split into two branches Q and R at any point (including the source) and converge again at a later point (including the sink), then one branch would get all the weight, whereas the other would be suppressed: the total contribution of the two branches is $X_\mathsf{Q} \sum u_\mathsf{Q} + (1 - X_\mathsf{Q}) \sum u_\mathsf{R}$, which attains its maximum at either $X_\mathsf{Q} = 1$ or $X_\mathsf{Q} = 0$.

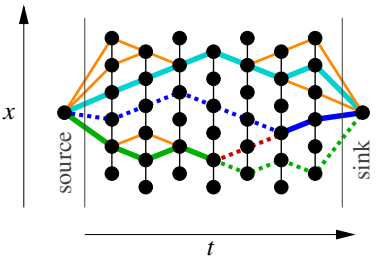

Fig. 1. LP-relaxation of multi-target tracking. On a junction-free path from source to sink (cyan), all variables are $X_{\ldots}=1$. Branching *within* a path is impossible (e.g. orange paths must have $X_{\ldots}=0$). A bridge (red) which permits to shift weight from one path (green) to another (blue) may cause non-integer values in the dashed regions.

– the branching argument applies recursively, so non-integer values can only occur when *two different paths* are connected by a "bridge", so that weight can be shifted from one to the other when their relative likelihood changes.

The solution $\mathbf{X}^*_{ILP}$ of the ILP is the maximum a-posteriori set of trajectories over the observed time window Φ. In practice this time interval is bounded by the available storage and computation power. The number M of variables and constraints to be stored grows linearly with Φ, and the average-case computational complexity of LP-solvers is $\mathcal{O}(M)$, too. A practical solution is to solve Eq. (1) for overlapping time intervals and constrain the solutions to be consistent by fixing the first frame. Empirically, intervals of $\Phi=30$ frames are sufficient.

3.2 Observation Model

We formulate tracking in world coordinates for the general case of multiple cameras observing the scene from different viewpoints. Multi-camera setups greatly improve tracking accuracy when the camera positions are low over the ground, such that one has to accept inaccurate depth estimates as well as frequent occlusions. The framework includes single-view tracking as a special case, by setting the number of cameras to 1. As usual, the posterior is split into an observation likelihood and a motion prior. We further decompose the observation into two parts measuring object detection response, respectively colour similarity:

$$P(X^t_{ijk}=1|\mathbf{R}) \sim P_O(\mathbf{R}|X^t_{ijk}=1) \cdot P_A(\mathbf{R}|X^t_{ijk}=1) \cdot P(X_{ijk}=1)\ . \qquad (5)$$

Object detection. To measure the support of targets in the image data, we use our own implementation of the popular HOG detector [13]. The detector scans the images I^t_ν (taken from viewpoints $\mathbf{c}_\nu$ at all three frames of the tracklet) over all positions $\mathbf{u}$ and scales s with a binary classifier trained to discriminate people from background, and returns for every location and scale a classification score R^t_ν. The scores are mapped from image locations $(\mathbf{u}, s)$ to locations $\mathbf{x}$ and target heights h in the world coordinate system with appropriate projections, and aggregated over all views and the three frames to obtain the total evidence $\mathbf{R}$ for a tracklet.

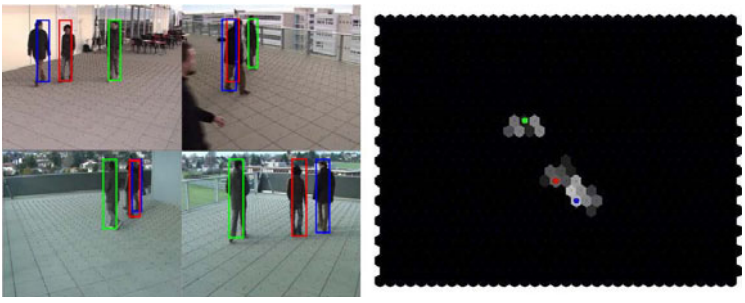

Fig. 2. The evidence $P(\mathbf{R}|X^t_{ijk})$ has smooth peaks, which are not precisely localised. (left) tracking results in four views. (right) birds-eye view of the scene. Note that the correct position for the green subject is *not* the one with the highest score. Our algorithm avoids per-frame decisions and chooses the best location during tracking.

The evidence at this point depends not only on X^t_{ijk}, but also on the person height h, via the detection scale s. In principle one could track directly in the $(\mathbf{x}, h)$-space, with a constraint that the height of any given person should not change over time. To reduce the computational burden, we prefer to place a Gaussian prior $P(h) = \mathcal{N}(h; \bar{h}, \sigma_h)$ on the person height and marginalise it out,

$$P_O(\mathbf{R}|X^t_{ijk}=1) = \sum_q \left(P_O(\mathbf{R}|X^t_{ijk}=1, h_q) \cdot P(h_q)\right) . \tag{6}$$

Appearance. The generic object model is complemented with a target-specific appearance model to better distinguish different targets. To this end, we demand that the colour distribution of a target varies slowly over short time spans. All sites of a tracklet X^t_{ijk} are projected back to the respective image locations $\mathbf{u}$, and at each location a colour histogram is extracted. The histograms of consecutive sites in a tracklet are then compared with the Bhattacharyya distance d_B, and the results are combined over all pairs of sites and all viewpoints $\mathbf{c}_\nu$:

$$P_A(\mathbf{R}|X^t_{ijk}=1) \sim \prod_{\mathbf{c}_\nu} \exp\left(-\frac{d_B(\mathbf{u}_i^{t-1}, \mathbf{u}_j^t) + d_B(\mathbf{u}_j^t, \mathbf{u}_k^{t+1})}{\sigma_B^2}\right) \tag{7}$$

3.3 Exclusion Constraints

Exclusion constraints between different tracklets ensure plausible interactions between the targets. The simplest form of constraint, which has been widely used in multi-target tracking, is the *collision avoidance* implemented by Eq. (3). We argue that exclusion constraints can also be applied over larger neighbourhoods, to incorporate non-maximum suppression (NMS) in the tracking framework rather than do it at the frame level, such that the retained location is the one which is optimal for the entire time interval, rather than for a single frame.

A main limitation of most tracking schemes is that non-maxima suppression is carried out on a per-frame basis. The evidence $P(\mathbf{R}|X^t_{ijk})$ measured by the observation model is in practice not a set of perfect spikes, but a smooth distribution

with peaks which are not well localised, see Fig. 2. To remedy this, the distribution is replaced by the modes only, found by some mode-seeking procedure like mean-shift or morphological erosion. Traditional non-maxima suppression thus commits to a location without taking into account the fact that target locations should be consistent over time. Instead, we propose to integrate NMS into tracking, rather than detection: the detector output is left to be ambiguous around the modes, and the optimisation can choose which location is most likely, given also evidence from neighbouring frames and the dynamic model. However, in this context an additional difficulty arises. Since the number of targets is not known a-priori, the evidence for a target at its neighboring locations can still be strong enough to generate multiple tracks. In other words, a prior is required, which formalises the intuition that plaits of intertwined trajectories are unlikely. To this end, we introduce a number of additional constraints, which prohibit not only collisions of targets at the *same* location, but also tracklets starting at immediately neighbouring locations (which amounts to the assumption that the grid sampling distance is smaller than the minimal possible distance between two targets),

$$\forall\, ijk \in \mathcal{T}, t : r \in \mathcal{S}(i) \Rightarrow X^t_{ijk} + X^t_{rjk} \leq 1 \tag{8}$$

These constraints prevent targets from moving too close to each other, and also avoid trajectories crossing in such a way that a collision would happen in the empty space between two grid locations.

We point out that the effect of the prior is not the same as single-frame NMS: under the exclusion constraints the optimisation is free to choose a target location $\mathbf{x}^t$, which is *not* a maximum of the detection score in frame t, in order to achieve a smoother trajectory, or to avoid collisions with other targets.

3.4 Dynamic Model

An important ingredient of tracking is the dynamic model, which encodes prior knowledge about likely motion patterns of the tracked objects. Using such dynamic models – mostly assuming constant heading, constant velocity or constant acceleration – has a long and successful tradition, however such models have been dropped in grid-based tracking.

To overcome this, we extend the grid-based formulation to incorporate the *constant heading* model, i.e. we assume that objects tend not to change their motion *direction*. A prerequisite for the ILP formulation is that the objective function Eq. (1) be linear. To preserve the linearity, the motion prior $P(X^t_{ijk} = 1)$ must be formulated such that it can be computed *locally for each variable* (i.e. its contribution must be part of the unary terms). This is the reason why we have introduced the *tracklets*: checking for constant heading requires two consecutive motion vectors, and hence three consecutive sites, thus the variables must cover at least three consecutive frames.

Given the two motion vectors $\mathbf{m}_{ij} = (x_j – x_i, y_j – y_i, 1)^\top$ and $\mathbf{m}_{jk} = (x_k – x_j, y_k – y_j, 1)^\top$ in a tracklet X^t_{ijk}, one can include the prior by penalising the heading change α between them, measured in (x, y, t)-space. The tracklet is assigned a

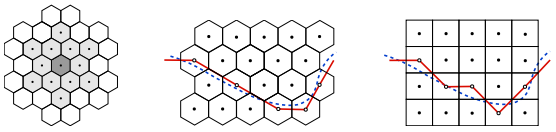

Fig. 3. (left) The 12-neighbourhood and symmetry axes in a hexagonal tiling. (middle, right) Aliasing of an example trajectory on a hexagonal grid and a rectangular grid with the same sample density.

probability which grows inversely with $\alpha^2(X^t_{ijk})$, such that deviations from the constant-heading assumption are penalised, as desired:

$$P(X^t_{ijk}=1) \sim \exp\big(-\tfrac{\alpha^2}{\sigma^2_\alpha}\big) \quad \text{where} \quad \alpha = \arccos \tfrac{\mathbf{m}_{ij}^\top \mathbf{m}_{jk}}{|\mathbf{m}_{ij}||\mathbf{m}_{jk}|} \tag{9}$$

Note that the angle α is computed in (x, y, t)-space. The method can be trivially extended to favour constant *velocity* by penalising the difference between $\mathbf{m}_{ij}$ and $\mathbf{m}_{jk}$, however we found the angle to work better, probably because of the varying step-length on a discrete grid.

The obvious effect of the dynamic model is that smoother, more accurate trajectories are estimated in the presence of inaccurate or weak evidence. Beyond its original purpose, the dynamic model also has a more subtle benefit on the optimisation: by penalising tracklets with strong heading changes, the motion prior sharpens the posterior, and thus the objective function U. As a consequence, the relaxation gap narrows, and fewer non-integer values occur. This effect is particularly strong in difficult circumstances, when the evidence $P(\mathbf{R}|X^t_{ijk}=1)$ is rather flat, such that the potential target locations spread out over a large number of tracklets. Therefore the dynamic model drastically reduces computation time (in our experiments by at least a factor of 10). In some cases the number of non-integer values without motion prior even becomes so high that it is no longer tractable to find an integral solution with branch-and-cut or probing.

3.5 Hexagonal Discretisation

To make tracking amenable to global optimisation with ILP, in the spirit of [8,1], the location space $\mathbf{x}$ must be discretised to a finite set of locations. As explained above, we prefer not to heuristically prune the per-frame likelihood $P(X^t_{ijk}|\mathbf{R})$ to a small set of permissible locations, but rather sample the ground plane in a regular lattice. A natural choice, which has been used in previous work, is a rectilinear grid, similar to the image grid. Unfortunately, such a grid has a strong preference for the two canonical directions along the x- and y-axes, whereas target trajectories in other directions exhibit severe aliasing.

Aliasing is not a big problem in the absence of a dynamic model, but together with the proposed motion model it creates difficulties: to check the deviation from constant heading *locally* one needs to rely on the vectors between the grid locations, thereby penalising trajectories which are not grid-aligned and hence continuously change directions. To alleviate this effect and boost the positive

effect of the dynamic model, we propose to use instead a hexagonal tiling of the ground plane, inducing a tri-axial neighbourhood system. In this grid, the 8-neighbourhood is replaced by a 12-neighbourhood, which reduces staircasing artifacts, and allows one to better enforce the constant heading assumption, see Fig. 3. The hexagonal tiling has been used in other contexts in image processing and computer vision [18,19], precisely because it has more preferred directions and reduces aliasing artifacts. Note that the change of sampling grid does not impair data quality: the transformation is performed when mapping the target probabilities from images to the world coordinate system, so there is no additional resampling step that would further blur the data.

4 Experiments

We present experiments on five different public multi-view video sequences. Sequences *campus-1* and *campus-2* [11] were both recorded from 3 different camera viewpoints, and have 2000, respectively 1400 frames showing up to 6 people moving outdoors. Sequences *terrace-1* and *terrace-2* [20] were both recorded from 4 viewpoints, and have 2000 frames each with up to 6 people, also moving freely outdoors. Finally, as a benchmark for monocular tracking we use sequence *PETS-S2L1* from the VS-PETS 2009 benchmark. The sequence is better suited for single-view tracking because of the elevated viewpoint. There are 795 frames showing up to 8 people moving in a street. The entire dataset contains 52 individual trajectories, which were manually annotated and used as ground truth. Due to the low target speed, we processed only every other frame of *PETS-S2L1* and every 6^{th} frame in the remaining four sequences, such that targets move approximately one grid unit from one frame to the next.

All experiments have been carried out with the same set of parameters. The two free parameters of our method are the standard deviations σ_α and σ_B, which govern the relative influence of detection score, colour similarity, and dynamic model (c.f. Sec. 3.2 and 3.4). To keep the optimisation tractable for long sequences, we follow the usual strategy and process overlapping time windows. This adds two further parameters, the number of frames Φ per window, and the overlap Ω. We set $\Phi = 30$ (when processing every 6^{th} frame at 25 fps, this amounts to ≈ 7 seconds) and $\Omega = 10$.

Figure 4 shows example results. Targets are tracked successfully over many frames, new targets entering the scene are initialised automatically. Especially the second example shows many targets moving in a small space. People are often occluded simultaneously in several views. Long-term occlusion is a main cause of failure, such as for the person marked in cyan.

In the *PETS-L2S1* sequence, up to 7 targets are tracked in *monocular* video over a large area of interest. Note the false positive on the tripod near the image centre: false detections on background objects are the dominant cause of false positives, since they tend to appear frequently on the same structures and, being static, fulfil the constraints of the dynamic model.

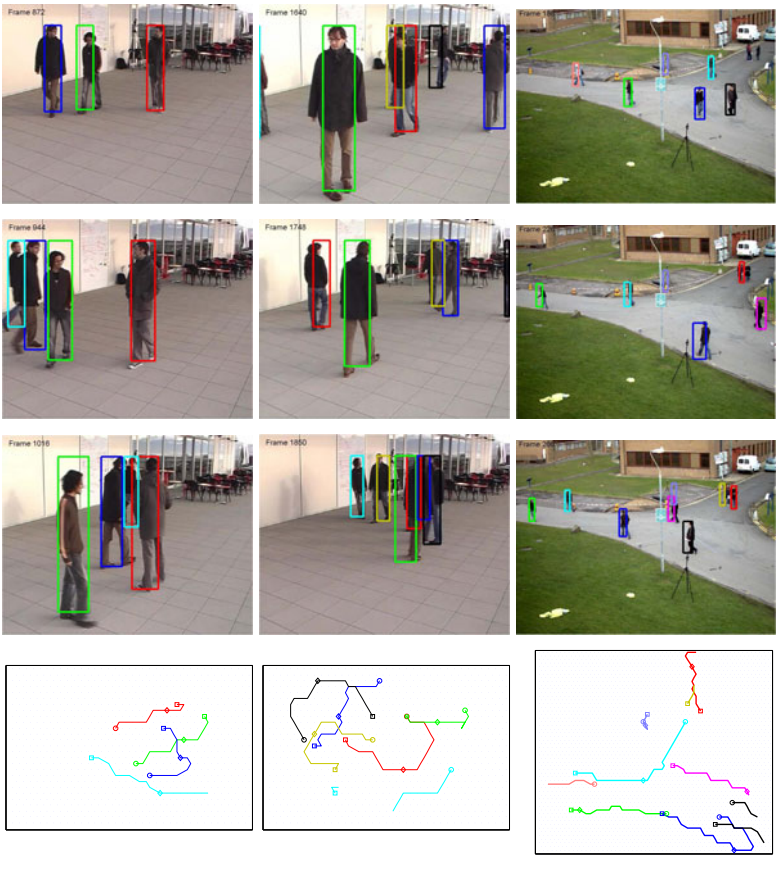

Fig. 4. Tracking results obtained with our algorithm. The left and middle column are from the *terrace-1* sequence, the right column is from *PETS-L2S1*. Displayed are three sample frames (1st-3rd row), and a birds-eye view of target trajectories (last row). The displayed frames are marked (top ◯, middle ♢, bottom □). See text for details.

4.1 Comparison to Previous Work

We directly compare the trajectories estimated by our method to those of [1], which we extracted from their published results. Their method is based on a similar ILP formulation, but on a rectilinear grid without dynamic model. Fig. 5 shows sample trajectories from both methods, with similar grid resolutions. The examples illustrate how late non-maxima suppression, together with the dynamic model, avoids implausible jittering. We emphasise that the improvement is due to the combination of all modelling choices: late non-maxima suppression preserves the necessary evidence for flexible target placement, while the dynamic model on a hexagonal grid supplies the constraints to handle the extra flexibility.

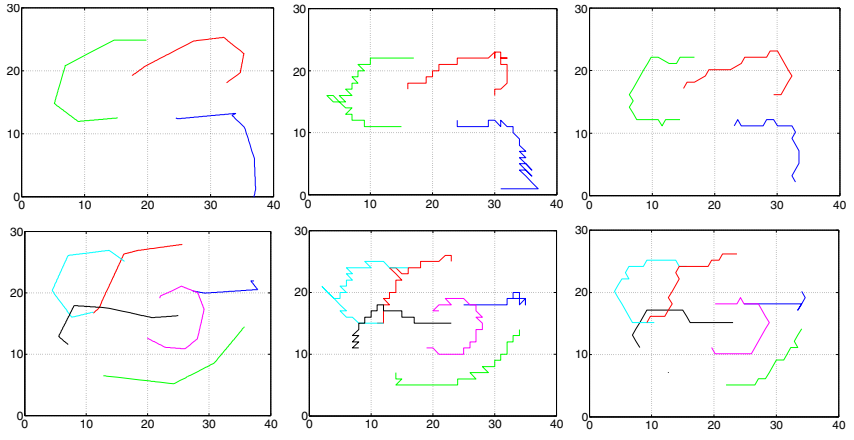

Fig. 5. Improved trajectories with the proposed model. *(left)* manually annotated ground truth for 200 frames of sequence *terrace-1*. *(middle)* trajectories reconstructed by state-of-the-art tracking *without* dynamic model [1]. *(right)* trajectories estimated by our system with dynamic model on a hexagonal grid.

4.2 Quantitative Evaluation

In the following we quantitatively evaluate our tracker against the baseline ILP tracker without dynamic model and operating on a rectilinear grid with either the standard 9-neighbourhood (8 neighbours and the central location itself) or a larger 21-neighbourhood. We use several metrics: on one hand we compute the CLEAR metrics for multi-object tracking [21] because of their growing popularity; on the other hand we count the number of trajectory fragments and ID switches, similar to [22]. Finally, we also measure the smoothness of the estimated trajectories by the average angle between the segments of all tracklets.

The evaluation results are summarised in Figure 6. As expected our model greatly improves trajectory smoothness, with a three- to five-fold reduction in the average tracklet angle for both multi-view and monocular tracking. The smoother trajectories also improve tracking accuracy: CLEAR-MOTA (measuring false negatives, false positives, and miss-matches) increases by 10-20%, because our model mitigates the effect of inaccurate and uncertain evidence. Using 21 instead of 9 neighbours also improves accuracy, but is still inferior to our result, while taking ≈ 5 times longer to compute due to the larger number of variables. Tracking precision (CLEAR-MOTP, measuring overlap of bounding boxes) improves insignificantly, because the metric is dominated by the alignment error due to the discrete location grid. At the same time, there is a dramatic reduction of fragmented tracks and identity switches ($\approx 50\%$ for the monocular case, 80-90% for the multi-view case). Trajectory fragments are generated when the tracker drifts away from a target, which is less likely if late non-maxima suppression and the motion prior can correct inaccuracies of the evidence. ID switches happen when data association fails for targets very close to each other. The

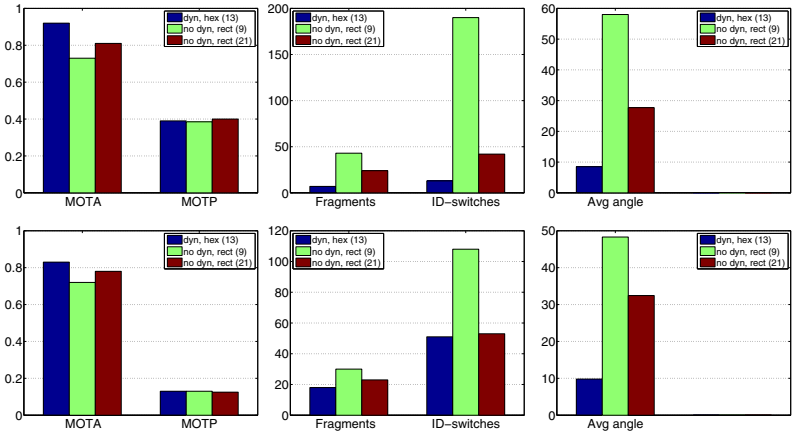

Fig. 6. Tracking performance. *(left)* CLEAR metrics – higher is better. *(middle)* fragmentation and ID switches – lower is better. *(right)* smoothness – lower is better. Globally optimal tracking benefits significantly from dynamic models on the hexagonal grid, both in multi-view (top row) and in the monocular setting (bottom row).

motion prior improves correct data association, because it favours the option with more plausible dynamics.

5 Conclusion

We have presented an algorithm for tracking a varying number of targets on a discrete location grid. Multi-target tracking is cast as integer linear programming, and solved through LP-relaxation, in most cases to global optimality. Compared to previous research in this direction, we have argued that tracking should use the original target evidence as input and perform non-maxima suppression during trajectory estimation, and we have demonstrated how to include standard dynamic models in the ILP formulation. We have also shown that best results are achieved on a hexagonal rather than a rectilinear grid.

The experimental comparison on public benchmark videos confirms that beyond its theoretical appeal the proposed formulation delivers better tracking results and achieves superior performance in quantitative comparisons.

In future work we plan to analyse the cases, in which the relaxation alone does not return a global optimum. We believe that the problem structure can be exploited to solve those cases more efficiently. We also plan to use the result of the method as initialisation for a continuous optimisation scheme to overcome limitations due to the restriction to the discrete grid.

References

1. Berclaz, J., Fleuret, F., Fua, P.: Multiple object tracking using flow linear programming. In: Winter-PETS (2009)

2. Black, J., Ellis, T., Rosin, P.: Multi-view image surveillance and tracking. In: Workshop on Motion and Video Computing (2002)
3. Mittal, A., Davis, L.: M^2Tracker: a multi-view approach to segmenting and tracking people in a cluttered scene. Int. J. Comput. Vision 51, 189–203 (2003)
4. Vermaak, J., Doucet, A., Perez, P.: Maintaining multimodality through mixture tracking. In: ICCV (2003)
5. Giebel, J., Gavrila, D., Schnörr, C.: A Bayesian framework for multi-cue 3rd object tracking. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3024, pp. 241–252. Springer, Heidelberg (2004)
6. Okuma, K., Taleghani, A., de Freitas, N., Little, J., Lowe, D.: A boosted particle filter: Multitarget detection and tracking. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 28–39. Springer, Heidelberg (2004)
7. Leibe, B., Schindler, K., Van Gool, L.: Coupled detection and trajectory estimation for multi-object tracking. In: ICCV (2007)
8. Jiang, H., Fels, S., Little, J.J.: A linear programming approach for multiple object tracking. In: CVPR (2007)
9. Ess, A., Leibe, B., Schindler, K., Van Gool, L.: A mobile vision system for robust multi-person tracking. In: CVPR (2008)
10. Zhang, L., Li, Y., Nevatia, R.: Global data association for multi-object tracking using network flows. In: CVPR (2008)
11. Berclaz, J., Fleuret, F., Fua, P.: Robust people tracking with global trajectory optimization. In: CVPR (2006)
12. Viola, P., Jones, M., Snow, D.: Detecting pedestrians using patterns of motion and appearance. Int. J. Comput. Vision 63, 153–161 (2005)
13. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
14. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: CVPR (1999)
15. Boros, E., Hammer, P.L.: Pseudo-boolean optimization. Discrete Appl. Math. 123, 155–225 (2002)
16. Wolf, J.K., Viterbi, A.M., Dixson, G.S.: Finding the best set of K paths through a trellis with application to multitarget tracking. IEEE T. Aero. Elec. Sys. 25 (1989)
17. Rother, C., Kolmogorov, V., Lempitsky, V.S., Szummer, M.: Optimizing binary mrfs via extended roof duality. In: CVPR (2007)
18. Miller, E.: Alternative tilings for improved surface area estimates by local counting algorithms. Comput. Vis Image Und. 74, 193–211 (1999)
19. Middleton, L., Sivaswamy, J.: Hexagonal Image Processing: A Practical Approach. Springer, Heidelberg (2005)
20. Fleuret, F., Berclaz, J., Lengagne, R., Fua, P.: Multi-camera people tracking with a probabilistic occupancy map. IEEE T. Pattern Anal. 30, 267–282 (2008)
21. Kasturi, R., Goldgof, D.B., Soundararajan, P., Manohar, V., Garofolo, J.S., Bowers, R., Boonstra, M., Korzhova, V.N., Zhang, J.: Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. IEEE T. Pattern Anal. 31, 319–336 (2009)
22. Li, Y., Huang, C., Nevatia, R.: Learning to associate: HybridBoosted multi-target tracker for crowded scene. In: CVPR (2009)