Coupled Detection and Trajectory Estimation for Multi-Object Tracking

Bastian Leibe[1] Konrad Schindler[1] Luc Van Gool[1,2]

[1]ETH Zurich, Switzerland [2]KU Leuven, Belgium

{leibe,konrads}@vision.ee.ethz.ch vangool@esat.kuleuven.be

Abstract

We present a novel approach for multi-object tracking which considers object detection and spacetime trajectory estimation as a coupled optimization problem. It is formulated in a hypothesis selection framework and builds upon a state-of-the-art pedestrian detector. At each time instant, it searches for the globally optimal set of spacetime trajectories which provides the best explanation for the current image and for all evidence collected so far, while satisfying the constraints that no two objects may occupy the same physical space, nor explain the same image pixels at any point in time. Successful trajectory hypotheses are fed back to guide object detection in future frames. The optimization procedure is kept efficient through incremental computation and conservative hypothesis pruning. The resulting approach can initialize automatically and track a large and varying number of persons over long periods and through complex scenes with clutter, occlusions, and large-scale background changes. Also, the global optimization framework allows our system to recover from mismatches and temporarily lost tracks. We demonstrate the feasibility of the proposed approach on several challenging video sequences.

1. Introduction

Monocular multi-object tracking is a challenging, but practically important problem. The task is to estimate multiple interacting object trajectories from a single image sequence, either in the 2D image plane or in 3D object space. Typically, tracking is modeled as some kind of first-order Markov chain, *i.e.* object locations at a time step t are predicted from those at the previous time step $(t-1)$ and then refined by comparing the object models to the current image data, whereupon the object models are updated and the procedure is repeated for the next time step. The Markov paradigm implies that trackers cannot recover from failure, since once they have lost track, the information handed on to the next time step is wrong. This is a particular problem in a multi-object scenario, where object-object interactions and occlusions are likely to occur.

A first step to address this limitation is to combine tracking with detection. This has only recently become feasible due to the rapid progress of object (class) detection [4, 14, 21, 22]. The idea is to run an object detector, trained either offline to detect an entire object category or online to detect specific objects [1, 8]. Its output can then constrain the trajectory search to promising image regions and serve to re-initialize in case of failure. Going one step further, one can directly use the detector output as data source for tracking (instead of *e.g.* color information).

Still, data association remains a difficult problem in multi-object tracking scenarios with many similar and mutually occluding targets. Classic multi-target trackers such as Multi-Hypothesis Tracking (MHT) [18] and Joint Probabilistic Data Association Filters (JPDAFs) [6] jointly consider the data association from sensor measurements to multiple overlapping tracks. While not restricted to Markov chains, they can only keep few time steps in memory due to the exponential task complexity. Moreover, originally developed for point targets, they generally do not take physical exclusion constraints between object volumes into account.

The aim of this work is to improve robustness in multi-object tracking by coupling object detection and tracking in a non-Markovian hypothesis selection framework. Our approach implements a feedback loop, which passes on predicted object locations as a prior to influence detection, while at the same time choosing between and reevaluating trajectory hypotheses in the light of new evidence. In contrast to previous approaches, which optimize individual trajectories in a temporal window [2, 23] or over sensor gaps [11], our approach tries to find a globally optimal combined solution for all detections and trajectories, while incorporating physical constraints such that no two objects can occupy the same physical space, nor explain the same image pixels at the same time. The task complexity is reduced by only selecting between a limited set of plausible hypotheses, which makes the approach computationally feasible.

The paper is structured as follows. After discussing related work, Section 2 presents our hypothesis selection framework integrating object detection and trajectory estimation. Sections 3 and 4 describe the baseline systems we employ for each of those two components, after which Section 5 introduces our coupled formulation and Section 6 discusses details of its implementation. Section 7 finally presents experimental results.

978-1-4244-1631-8/07/$25.00 ©2007 IEEE

Related Work. In this paper, we address multi-object tracking in a surveillance scenario with a single, calibrated camera. Tracking in such a scenario consists of two subproblems: trajectory initialization and target following. While many approaches rely on background subtraction from a static camera for the former (*e.g.* [20, 12, 2]), several recent approaches have started to explore the possibilities of combining tracking with detection [17, 1, 8, 22]. This has been helped by the astonishing progress object detection research has made over the last few years [4, 14, 16, 21], which has resulted in state-of-the-art detectors that are applicable in complex outdoor scenes.

The second subproblem is typically addressed by classic tracking approaches, such as Extended Kalman Filters (EKF) [7], particle filtering [10], or Mean-Shift tracking [3], which rely on a Markov assumption and carry the associated danger of drifting away from the correct target. This danger can be reduced by optimizing data assignment and considering information over several time steps, as in MHT [18] and JPDAF [6]. However, task complexity limits previous optimization approaches to consider either only few time steps [18] or only single trajectories over longer time windows [2, 11, 23]. In contrast, our approach simultaneously optimizes detection and trajectory estimation for multiple interacting objects and over long time windows by operating in a hypothesis selection framework.

2. Approach

MDL Hypothesis Selection. Our basic mathematical tool is a model selection framework as introduced in [15]. We briefly repeat its general form here and later explain specific versions for object detection and trajectory estimation.

The intuition of the method is that in order to correctly handle the interactions between multiple models required to describe a data set, one cannot fit them sequentially (because interactions with models which have not yet been estimated would be neglected). Instead, an over-complete set of hypothetical models is generated, and the best subset selected with model selection in the spirit of the minimum description length (MDL) criterion.

To select the best models, the *savings* (in coding length) of each hypothesis h are expressed as

$$S_h \sim S_{data} - \kappa_1 S_{model} - \kappa_2 S_{error} \ , \qquad (1)$$

where S_{data} corresponds to the number N of data points, which are explained by h; S_{model} denotes the cost of coding the model itself; S_{error} describes the cost for the error committed by the representation; and κ_1, κ_2 are constants to weight the different factors. If the error term is chosen as the log-likelihood over all data points x assigned to a hypothesis h, then the following approximation holds[1]:

$$\begin{aligned} S_{error} &= -\log \prod_{x \in h} p(x|h) = -\sum_{x \in h} \log p(x|h) \qquad (2) \\ &= \sum_{x \in h} \sum_{n=1}^{\infty} \frac{1}{n} (1 - p(x|h))^n \approx N - \sum_{x \in h} p(x|h). \end{aligned}$$

Substituting eq.(2) into eq.(1) yields an expression for the merit of model h:

$$S_h \sim -\kappa_1 S_{model} + \sum_{x \in h} \left((1 - \kappa_2) + \kappa_2 p(x|h) \right) \ . \qquad (3)$$

Essentially, the merit of a putative model is the sum over its data assignment likelihoods, regularized with a term which compensates for unequal sampling of the data.

A data point can only be assigned to one model. Hence, overlapping hypothetical models compete for data points. This competition translates to interaction costs, which apply only if both hypotheses are selected. [15] has shown that the optimal set of models in such a scenario is given by the solution of the Quadratic Boolean Problem (QBP)

$$\max_n n^{\mathsf{T}} S n \quad , \quad S = \begin{bmatrix} s_{11} & \cdots & s_{1N} \\ \vdots & \ddots & \vdots \\ s_{N_1} & \cdots & s_{NN} \end{bmatrix} \ . \qquad (4)$$

Here, $n = [n_1, n_2, \ldots, n_N]^{\mathsf{T}}$ is a vector of indicator variables, such that $n_i = 1$ if hypothesis h_i is accepted, and $n_i = 0$ otherwise. S is an interaction matrix, whose diagonal elements s_{ii} are the merit terms (3) of individual hypotheses, while the off-diagonal elements $(s_{ij} + s_{ji})$ express the interaction costs between two hypotheses h_i and h_j. This formulation thus provides a mathematical tool to handle pairwise interactions between hypotheses.

Object Detection. For object detection, we use the pedestrian detector of [14], which utilizes the model selection framework explained above. A full description is beyond the scope of this paper. In a nutshell, a voting scheme based on multi-scale interest points generates a large number of hypothetical detections. From this redundant set, the subset with the highest joint likelihood is selected by maximizing $n^{\mathsf{T}} S n$: the binary vector n indicates which detection hypotheses shall be used to explain the image observations and which ones can be discarded. The interaction matrix S contains the hypotheses' individual savings, as well as their interaction costs, which assure that each image pixel is part of at most one detection (details are given in Section 3).

Trajectory estimation. In [13], a similar formalism is also applied to estimate object trajectories over the ground plane. The image detections in a 3D spacetime volume are linked to hypothetical trajectories with a simple dynamic model, and the best set of trajectories is selected from those hypotheses by solving another maximization problem $m^{\mathsf{T}} Q m$, where the interaction matrix Q again contains the individual savings and the interaction costs which arise if two hypotheses compete to fill the same part of the spacetime volume (see Section 4).

[1] This approximation improves robustness against outliers by mitigating the non-linearity of the logarithm near 0, while providing good results for unambiguous point assignments.

Coupled Detection & Trajectory Estimation. As shown above, both object detection and trajectory estimation can be formulated as individual QBPs. However, the two tasks are closely coupled: the merit of a putative trajectory depends on the number and strength of the underlying detections $\{n_i = 1\}$, while the merit of a putative detection depends on the current object trajectories $\{m_i = 1\}$, which impose a prior on object locations. These dependencies lead to further interactions between detections and trajectories. In this paper, we therefore jointly optimize both detections and trajectories by coupling them in a *combined* QBP.

However, we have to keep in mind that the relationship between detections and trajectories is not symmetric: trajectories ultimately rely on detections to be propagated, but new detections can occur without a trajectory to assign them to (*e.g.* when a new object enters the scene). In addition to the index vectors m for trajectories and n for detections, we therefore need to introduce a list of virtual trajectories v, one for each detection in the current image, to enable detections to survive without contributing to an actual trajectory. We thus obtain the following joint optimization problem

$$\max_{m,v,n} \begin{bmatrix} m^\top & v^\top & n^\top \end{bmatrix} \begin{bmatrix} \widetilde{Q} & U & V \\ U^\top & R & W \\ V^\top & W^\top & \widetilde{S} \end{bmatrix} \begin{bmatrix} m \\ v \\ n \end{bmatrix}, \tag{5}$$

where the elements of V, W model the interactions between detections and real and virtual trajectories, respectively, and U models the mutual exclusion between the two groups. The solution of (5) jointly optimizes both the detection results for the current frame, given the trajectories of the tracked objects, and the trajectories across frames, given the detections. As will be explained in Section 5, we approximate the computationally expensive full solution by an EM-style iterative approximation.

3. 2D Object Detection

The object detection part relies on the pedestrian detector from [14] to deliver an initial set of detections. With the help of a camera calibration, the 2D detections are converted to 3D object locations H on the ground plane, and their score is expressed in terms of the pixels they occupy

$$\begin{aligned} p(H|I) \;&\sim\; p(I|H)p(H) \\ &= p(H)\prod_{\mathbf{p}\in I} p(\mathbf{p}|H) = p(H)\prod_{\mathbf{p}\in Seg(H)} p(\mathbf{p}{=}\mathit{fig.}|H), \end{aligned} \tag{6}$$

where $Seg(H)$ denotes the support region of H in the image I, as returned by the detector. The location prior $p(H)$ is split up into a uniform distance prior for the detector's target range and a Gaussian prior for typical pedestrian sizes $p(H_{size}) \sim \mathcal{N}(1.7, 0.2^2)$ [meters], similar to [9].

Two detections H_i and H_j interact if they compete for the same image pixels. In this case, we assume that the hypothesis $H_k \in \{H_i, H_j\}$ farthest from the camera is occluded and subtract its support in the overlapping image area. For notational convenience, we define the pseudo-likelihood

$$p^*(H|I) = \sum_{\mathbf{p}\in Seg(H)} \big((1-\kappa_2) + \kappa_2 p(\mathbf{p}{=}\mathit{fig.}|H)\big) + \log p(H) \tag{7}$$

and obtain, with the approximation from eq. (2), the following terms for the object detection matrix S:

$$\begin{aligned} s_{ii} &= -\kappa_1 + p^*(H_i|I) \\ s_{ij} &= -\frac{1}{2}\sum_{\mathbf{p}\in Seg(H_i\cap H_j)} \big((1-\kappa_2) + \kappa_2 p(\mathbf{p}{=}\mathit{fig.}|H_k)\big) + \kappa_2 \log p(H_k) \end{aligned} \tag{8}$$

For each detection, we additionally compute an object-specific color model a_i, represented by an $8{\times}8{\times}8$ RGB histogram, over the object detector's confidence region. This color model will later be used to help group consistent detections into trajectories.

4. Spacetime Trajectory Following

The optimization is fed hypothetical tracks based on the object detections in spacetime. It is this task of finding hypotheses which we mean by *trajectory following*, not the final selection of the best set of tracks. To find plausible hypotheses and to estimate their parameters, we use the Event Cone following framework from [13], which models individual trajectory hypotheses with the help of EKFs [7]. The idea is simple: start from a detection $H_{i,t}$, set the initial velocity to $v = 0$, and build up a trajectory by iteratively predicting new locations at adjacent timesteps, and updating them based on the actual detections found near the predicted location. By repeating this for different starting points, an initial set of putative trajectories is generated, which will then be fed into the hypothesis selection procedure.

Trajectory Search. Each trajectory $\mathcal{H}_{t_0:t}$ is represented by a dynamic model $\mathcal{D}$ and an appearance model $\mathcal{A}$. As dynamic model, we adopt linear prediction with two adaptive parameters, velocity and bearing. The positional uncertainty around the predicted location $[x^p_{t+1}, y^p_{t+1}]^\top$ is approximated by an oriented Gaussian. Note that the dynamics are modeled in the 3D ground plane, not in the image plane. The appearance model $\mathcal{A}$ is defined as the trajectory's color histogram, which evolves as the trajectory progresses.

With a threshold for the likelihood, the EKF defines a cone in spacetime. Given a partially grown trajectory $\mathcal{H}_{t_0:t}$, we search for candidate observations H_{i,t_i} that fall inside this cone and evaluate them under the trajectory's model for the current time step, weighted with a temporal discount λ:

$$\begin{aligned} p(H_{i,t_i}|\mathcal{H}_{t_0:t}) &= p(\mathcal{H}_{t_i}|\mathcal{H}_{t_0:t})p(H_{i,t_i}|\mathcal{H}_{t_i}) \\ &= e^{-\lambda(t-t_i)}p(H_{i,t_i}|\mathcal{A}_{t_i})p(H_{i,t_i}|\mathcal{D}_{t_i}). \end{aligned} \tag{9}$$

After this, the trajectory is updated by the weighted mean of its predicted position and the supporting observations:

$$\mathbf{x}_{t+1} = \frac{1}{Z}\left(p(\mathcal{H}_{t+1}|\mathcal{H}_{t_0:t})\mathbf{x}^p_{t+1} + \sum_i p(H_{i,t+1}|\mathcal{H}_{t_0:t})\mathbf{x}_i\right), \tag{10}$$

with $p(\mathcal{H}_{t+1}|\mathcal{H}_{t_0:t}) = e^{-\lambda}$, and normalization factor Z. Velocity, rotation, and appearance model are updated in the same fashion.

Merit and Interactions. We express the support $\mathcal{S}$ of a trajectory hypothesis $\mathcal{H}_{t_0:t}$ reaching from t_0 to t by the evidence collected from the images $I_{t_0:t}$ during that time span:

$$\begin{aligned}\mathcal{S}(\mathcal{H}_{t_0:t}|I_{t_0:t}) &= p(\mathcal{H}_{t_0:t})\sum_i \frac{p(H_{i,t_i}|\mathcal{H}_{t_0:t})}{p(H_{i,t_i})} p(H_{i,t_i}|I_{t_i}) \\ &\sim p(\mathcal{H}_{t_0:t})\sum_i p(H_{i,t_i}|\mathcal{H}_{t_0:t}) p(H_{i,t_i}|I_{t_i}), \quad (11)\end{aligned}$$

For trajectory estimation, this support is used to define the trajectory interaction matrix Q as follows:

$$\begin{aligned} q_{ii} &= -\epsilon_1 c(\mathcal{H}_{i,t_0:t}) + \sum_{H_{k,t_k}\in\mathcal{H}_i} ((1-\epsilon_2) + \epsilon_2\, g_{k,i}) \\ q_{ij} &= -\frac{1}{2} \sum_{H_{k,t_k}\in\mathcal{H}_i\cap\mathcal{H}_j} ((1-\epsilon_2) + \epsilon_2\, g_{k,*} + \epsilon_3\, O_{ij}), \quad (12) \\ g_{k,i} &= p^*(H_{k,t_k}|I_{t_k}) + \log p(H_{k,t_k}|\mathcal{H}_i), \end{aligned}$$

where $\mathcal{H}^* \in \{\mathcal{H}_i, \mathcal{H}_j\}$ denotes the weaker of the two trajectory hypotheses; $c(\mathcal{H}_{t_0:t})$ is a model cost that penalizes holes in the trajectory; and O_{ij} measures the physical overlap between the spacetime volumes of $\mathcal{H}_i$ and $\mathcal{H}_j$ given average object dimensions.

Thus, two overlapping trajectory hypotheses compete both for supporting observations and for the physical space they occupy during their lifetime. This makes it possible to model complex object-object interactions, such that two pedestrians cannot walk through each other or that one needs to yield if the other shoves.

5. Coupling between Detection and Tracking

Equations (6) and (11) define the support that is used to build up our coupled detection/tracking optimization problem. Because of its asymmetric nature, we however have to split up this support between the original matrices Q, S and the coupling matrices U, V, W. This is done as follows.

The modified interaction matrix $\widetilde{Q}$ for the real trajectories keeps the form from eq.(12), with the exception that only the support from previous frames is entered into $\widetilde{Q}$:

$$\begin{aligned} \widetilde{q}_{ii} &= -\epsilon_1 c(\mathcal{H}_{i,t_0:t}) + \sum_{H_{k,t_k}\in\mathcal{H}_{i,t_0:t-1}} ((1-\epsilon_2) + \epsilon_2\, g_{k,i}) \quad (13) \\ \widetilde{q}_{ij} &= -\frac{1}{2} \sum_{H_{k,t_k}\in(\mathcal{H}_i\cap\mathcal{H}_j)_{t_0:t-1}} ((1-\epsilon_2) + \epsilon_2\, g_{k,*} + \epsilon_3\, O_{ij}), \quad (14) \end{aligned}$$

The matrix R for the virtual trajectories contains simply the entries $r_{ii} = \varepsilon, r_{ij} = 0$, with ε a very small constant, and the matrix U for the interaction between real and virtual trajectories has entries u_{ik} which are computed similar to the real trajectory interactions q_{ij}

$$u_{ik} = -\frac{1}{2}((1-\epsilon_2) + \epsilon_2\, g_{k,i} + \epsilon_3\, O_{ik}). \quad (15)$$

$$\begin{bmatrix} \widetilde{Q} & U & V \\ U^\top & R & W \\ V^\top & W^\top & \widetilde{S} \end{bmatrix} =$$

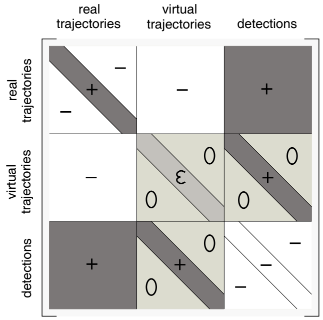

Figure 1. *Structure of the coupled optimization matrix (eq. (5)).*

The modified object detection matrix $\widetilde{S}$ contains as diagonal entries only the base cost of a detection, and as off-diagonal elements the full interaction cost between detections,

$$\widetilde{s}_{ii} = -\kappa_1\epsilon_2 - (1-\epsilon_2), \qquad \widetilde{s}_{ij} = s_{ij}. \quad (16)$$

Finally, the interaction matrices V, W between trajectories and detections have as entries the evidence a new detection contributes towards explaining the image data (which is the same as its contribution to a trajectory),

$$\begin{aligned} v_{ij} &= \frac{1}{2}((1-\epsilon_2) + \epsilon_2 p^*(H_j|I_t) + \epsilon_2 \log p(H_j|\mathcal{H}_i)) \quad (17) \\ w_{jj} &= \max_i [v_{ij}] \quad (18) \end{aligned}$$

Note that R, S, and W are all quadratic and of the same size $N \times N$ and that R and W are diagonal matrices. As can be easily verified, the elements of the submatrices indeed add up to the correct objective function. Figure 1 visualizes the structure of the completed optimization matrix.

To illustrate this definition, we describe the most important features of the coupled optimization problem in words: 1) A trajectory is selected if its score outweighs the base cost in $\widetilde{q}_{ii}$. 2) If trajectory $\mathcal{H}_i$ is selected, and a compatible detection H_j is also selected, then H_j contributes to the trajectory score through v_{ij}. 3) If a detection H_j is not part of any trajectory, but its score outweighs the base cost in $\widetilde{s}_{jj}$, then it is still selected, with the help of its virtual trajectory and the contribution w_{jj}. 4) If a detection is part of any selected trajectory, then its virtual trajectory will not be selected, due to the interaction costs u_{ij} and the fact that the merit r_{jj} of a virtual trajectory is less than that of any real trajectory. 5) Finally, while all this happens, the detections compete for pixels in the image plane through the interaction costs $\widetilde{s}_{ij}$, and the trajectories compete for space in the object coordinate system through $\widetilde{q}_{ij}$.

Recapitulating the above, coupling has the following effects. First, it supports novel object detections that are consistent with existing trajectories. Eq. (17) states that existing trajectories impose a prior $p(H_j|\mathcal{H}_i)$ on certain object locations which raises the chance of generating novel detections there above the uniform background level $\mathcal{U}$. We model this prior as a Gaussian around the projected object position using the trajectory's dynamic model $\mathcal{D}$, so that

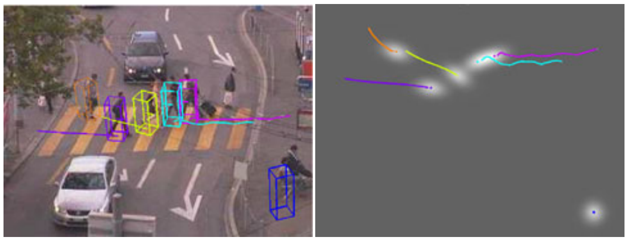

Figure 2. *Influence of past trajectories on object detection.* Left: *25th frame of sequence 2, and detected pedestrians.* Right: *Illustration of the detection prior for the 26th frame. Top view showing trajectories estimated in the last frame, predicted positions, and detection prior (brighter color means higher probability).*

$p(H_j|\{\mathcal{H}_i\}) = \max[\mathcal{U}, \max_i[\mathcal{N}(\mathbf{x}_i^p, \sigma^2_{pred})]]$. Fig. 2 shows the prior for a frame from one of our test sequences. Second, the evidence from novel detections aids trajectories with which those detections are consistent by allowing them to account the new information as support.

Iterative Optimization. Optimizing eq. (5) directly is difficult, since quadratic boolean optimization in its general form is NP hard. However, many QBPs obey additional simplifying constraints. In particular, the hypothesis selection problems for Q and S described earlier are submodular, and the expected solution is sparse (only few hypotheses will be selected), which allows one to find strong local maxima, as shown in [19]. However, the new QBP (5) is no longer submodular, since the interaction matrices V and W have positive entries.

We therefore resort to an EM-style iterative solution, which lends itself to the incremental nature of tracking: at each time step t, object detection is solved using the trajectories from the previous frame $(t-1)$ as prior. In the above formulation, this corresponds to fixing the vector m. As an immediate consequence, we can split the detection hypotheses into two groups: those which are supported by a trajectory, and those which are not. We will denote the former by another binary index vector n^+, and the latter by its complement n^-. Since for fixed m the term $m^\top Q m = const.$, selecting detections amounts to solving

$$\begin{aligned}&\max_{v,n}\left[\begin{bmatrix}v^\top & n^\top\end{bmatrix}\begin{bmatrix}R & W\\ W^\top & S\end{bmatrix}\begin{bmatrix}v\\ n\end{bmatrix}+2m^\top\begin{bmatrix}U & V\end{bmatrix}\begin{bmatrix}v\\ n\end{bmatrix}\right]=\\ &\max_{v,n}\begin{bmatrix}v^\top & n^\top\end{bmatrix}\begin{bmatrix}R+2\,\mathrm{diag}(U^\top m) & W\\ W^\top & S+2\,\mathrm{diag}(V^\top m)\end{bmatrix}\begin{bmatrix}v\\ n\end{bmatrix}.\end{aligned}\tag{19}$$

The interactions $U^\top m$ by construction only serve to suppress the virtual trajectories for the n^+. In contrast, $V^\top m$ adds the detection support from the n^+ to their score, while the diagonal interaction matrix W does the same for the n^-, which do not get their support through matrix V. We can hence further simplify to

$$\max_{n}\left[n^\top\left(R+S+2\,\mathrm{diag}(V^\top m)+2\,\mathrm{diag}(W^\top n^-)\right)n\right]. \tag{20}$$

The support W is only applied if no support comes from the trajectories and if in turn the interaction cost $U^\top m$ can be dropped, which only served to make sure W is outweighed for any n^+. The solution $\widehat{n}$ of (20) is the complete set of detections for the new frame; the corresponding virtual trajectories are $\widehat{n} \cap n^-$.

With the detection results from this step, the set of optimal trajectories is updated. This time, the detection results $[v^\top n^\top]$ are fixed, and the optimization reduces to

$$\max_{m}\left[m^\top\left(Q+2\,\mathrm{diag}(Vn)+2\,\mathrm{diag}(Uv)\right)m\right]. \tag{21}$$

The third term can be dropped, since virtual trajectories are now superseded by newly formed real trajectories. The second term is the contribution which the new detections make to the trajectory scores. The two reduced problems (20) and (21) are again submodular and can be solved with the multi-branch ascent method introduced in [19].

6. Detailed Implementation

An advantage of our approach is that, based on object detection, it can track both moving and static objects. This is a major limitation of tracking approaches based on background subtraction, which often integrate static objects into their dynamically updated background models [20]. However, for optimal results we have to treat the two hypothesis types differently. Object detector output is often subject to some jitter caused by noise in the camera signal. While a moving hypothesis will try to follow that jitter in order not to miss an upcoming motion of its target object, a static hypothesis should integrate detections over a longer time frame in order to arrive at stable localization. Separate procedures are therefore used for static and dynamic objects. In both cases, three processes are required: 1) to extend existing hypotheses; 2) to generate new ones; and 3) to prune away unsuccessful ones.

Static Objects. For a static object, the sequence of prediction cones collapses to a spacetime cylinder with constant radius. New static objects are initialized with simple mean-shift clustering on the groundplane: clusters of detections around a fixed groundplane location which exhibit sufficient continuity over time are regarded as potential locations of static objects. For efficiency, only detections from the few most recent frames serve as starting points. The cluster center defines the object location; a weighted mean over the cluster members its appearance model.

To avoid unnecessary computations, existing hypotheses are extended in each time step: among the detections from the current time step t, inliers to the existing cluster are found and added, and the cluster center and appearance model are updated by recomputing the new weighted mean. If no inliers are found, the trajectory is simply extrapolated through time, leaving the values unchanged.

Moving Objects. Similar mechanisms also serve to update the set of moving object hypotheses. New trajectories are

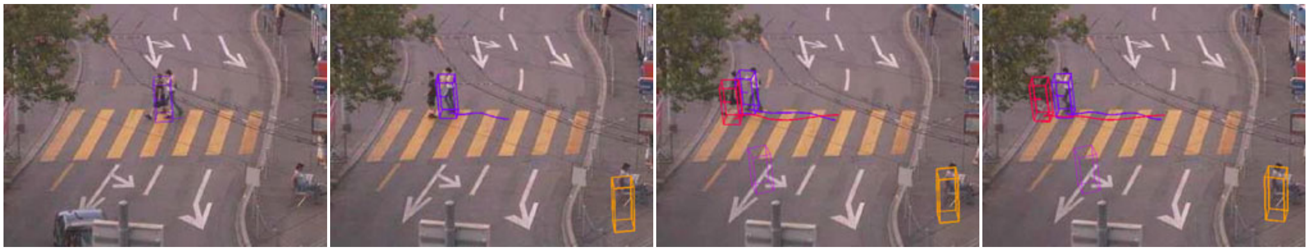

Figure 3. *Example tracking results visualizing the non-Markovian nature of our approach. At the beginning of the sequence, both pedestrians walk close together and only one trajectory is initialized. However, when they separate sufficiently, a second trajectory is added that reaches back to the moment when both were first observed, while the first trajectory is automatically adjusted to make room for it.*

initialized with the EKF framework detailed above: starting from detections in the last few frames, trajectory points are predicted both forward and backward in time, and their location and appearance models are updated by a weighted mean over the prediction and the detections within the uncertainty ellipse. With the new location, the dynamic model over the ground plane is updated, and a new location and uncertainty ellipse are predicted. Note that the procedure does not require a detection in every frame: the procedure's time horizon can be set to tolerate large temporal gaps.

Dynamic model propagation is unidirectional. After finding new evidence, the already existing part of the trajectory is not re-adjusted. However, in order to reduce the effect of localization errors, inevitably introduced by limitations of the object detector, the final trajectory hypothesis is smoothed by local averaging, and its score (11) is recomputed. Similar to the static case, repeated computations are avoided by extending hypotheses from previous time steps.

Hypothesis Pruning. Continually extending the existing hypotheses (while generating new ones) leads to an evergrowing hypothesis set, which would quickly become intractable. A conservative pruning procedure is used to control the number of hypotheses to be evaluated: candidates extrapolated through time for too long without finding any new evidence are removed. Similarly, candidates which have been in the hypothesis set for too long without having ever been selected are discontinued (these are mostly weaker hypotheses, which are always outmatched by others in the competition for space). Importantly, the pruning step only removes hypotheses which have been unsuccessful over a long period of time. All other hypotheses, including those not selected during optimization, are still propagated and are thus given a chance to find new support at a later point in time. This allows the tracker to recover from failure and retrospectively correct tracking errors.

Identity Management. The hypothesis selection framework helps to ensure that all available information is used at each time step. However, it delivers an *independent explanation* at each time step and hence does not by itself keep track of object identities. Frame-to-frame propagation of tracked object identities is a crucial capability of tracking (as opposed to frame-by-frame detection).

Propagating identity is trivial in the case where a trajectory has been generated by extending one from the previous frame, where the hypothesis ID is simply passed on, as in a recursive tracker. However, one of the core strengths of the presented approach is that it does not rely on stepwise trajectory extension alone. If at any time a newly generated hypothesis provides a better explanation for the observed evidence than an extended one, it will replace the older version. However, in this situation the new trajectory should inherit the old identity, in order to avoid an identity switch.

The problem can be solved with a simple heuristic based on the associated data points: the identities of all selected trajectories are written into a buffer, together with the corresponding set of explained detections. This set is continuously updated as the trajectories grow. Each time a new trajectory is selected for the first time, it is compared to the buffer, and if its set of explained detections is similar to an entry in the buffer, it is identified as the new representative of that ID, replacing the older entry. If it does not match any known trajectory, it is added to the buffer with a new ID.

Trajectory Initialization and Termination. Object detection yields fully automatic initialization. Given a new sequence, the system accumulates pedestrian detections in each new frame and tries to link them to detections from previous frames to obtain plausible spacetime trajectories, which are then fed into the selection procedure. After a few frames, the merit of a correct trajectory exceeds its cost, and an object track is started. Although several frames are required as evidence for a new track, the trajectory is in hindsight recovered from its beginning.

The automatic initialization however means that trajectory termination needs to be handled explicitly: if an object leaves the scene, the detections along its track still exist and may prompt unwanted re-initializations. To control this behavior, exit zones are defined in 3D space along the image borders and are constantly monitored. When an object's trajectory enters the exit zone from within the image, the object is labeled as terminated, and its final trajectory is stored in a list of terminated tracks. To keep the tracker from reusing the underlying data, all trajectories from the termination list are added to the trajectory set and are always se-

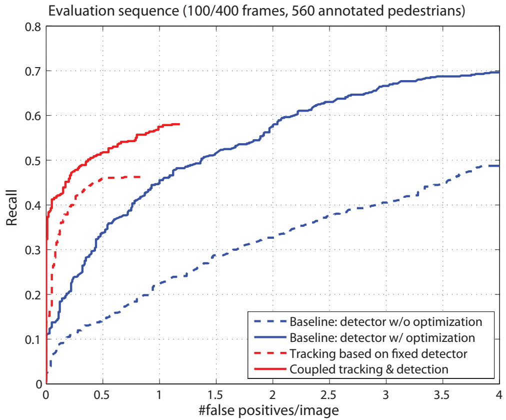

Figure 4. *Performance comparison of our coupled detection+tracking system compared to various baselines.*

lected, thus preventing re-initializations based on the same detections through their interaction costs. The list of terminated tracks effectively serves as a memory, which ensures that the constraint that no two objects can occupy the same physical space survives after a hypothesis' termination.

7. Experimental Results

We now present results on several challenging test sequences. All sequences were recorded with a public webcam at 15fps, 320×240 pixels resolution, and contain severe MPEG compression artifacts. In all result figures, line width denotes confidence of the recovered tracks: trajectories rendered with thin lines have lower scores.

Fig. 3 visualizes our approach's behavior on a short test sequence of two pedestrians crossing a street. First, they walk close together and the detector often yields only a single detection. Thus, the support only suffices for a single trajectory to be initialized. However, as soon as the pedestrians separate, a second trajectory is instantiated that reaches back to the point at which both pedestrians were first observed. Together, the two trajectories provide a better explanation for the accumulated evidence and are therefore preferred by model selection. As part of our optimization, both trajectories are automatically adjusted such that their spacetime volumes do not intersect.

A more challenging case is displayed in Fig. 5. Here, multiple people cross the street at the same time, meeting in the middle. It can be seen that, caused by the occlusion, our system temporarily loses track of two pedestrians, resulting in identity switches. However, it automatically recovers after few frames and returns to the correct identities. Again, this is something that classical Markovian tracking approaches are unable to do. In addition, our approach is able to detect and track the sitting person in the lower right corner which is indistinguishable from a static background. Relying on an object detector for input, we are however limited by the quality of the detections the latter can provide. Thus, our system will hypothesize wrong tracks in locations where the detector consistently produces false alarms.

For a quantitative assessment, we annotated every 4th frame of this sequence manually. We marked all image locations with 2D bounding boxes in which a person was visible. We then derived similar bounding boxes from the tracked 3D volumes and compared them to the annotations. Following recent object detection evaluations, we consider a box as correct if it overlaps with the ground-truth annotation by more than 50% using the intersection-over-union criterion [5]. Only one bounding box per annotation is counted as correct; every additional one is counted as a false positive. Note that this compares only localization accuracy, not person identities. Fig. 4 shows the result of our coupled system, compared to the baseline delivered by the object detector (just matrix S) and to a baseline from a tracker based on fixed detections (decoupled matrices Q and S). Our approach improves on both baselines and results in increased localization precision.

Finally, Fig. 6 presents results on a very challenging sequence with large-scale background changes from an incoming tram, many static pedestrians, and frequent occlusions. The results confirm that our approach can deal with those difficulties and track its targets over long periods.

8. Conclusion

We have presented a novel approach for multi-object tracking that couples detection and trajectory estimation in a combined optimization problem. Our approach does not rely on a Markov assumption, but can integrate information over long time frames to revise its decision and recover from mistakes in the light of new evidence. Qualitative and quantitative results over several challenging test sequences demonstrate our method's performance.

Acknowledgments: This work is supported, in parts, by EU projects HERMES (IST-027110) and DIRAC (IST-027787).

References

[1] S. Avidan. Ensemble tracking. In *CVPR'05*, 2005.

[2] J. Berclaz, F. Fleuret, and P. Fua. Robust people tracking with global trajectory optimization. In *CVPR'06*, 2006.

[3] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *Trans. PAMI*, 25(5):564–575, 2003.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR'05*, 2005.

[5] M. Everingham and others (34 authors). The 2005 PASCAL Visual Object Class Challenge. In *Sel. Proc. of the 1st PASCAL Challenges Workshop*, LNAI. Springer, 2006.

[6] T. Fortmann, Y. Bar Shalom, and M. Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE J. Oceanic Engineering*, 8(3):173–184, 1983.

[7] A. Gelb. *Applied Optimal Estimation*. MIT Press, 1996.

[8] H. Grabner and H. Bischof. On-line boosting and vision. In *CVPR'06*, pages 260–267, 2006.

[9] D. Hoiem, A. Efros, and M. Hebert. Putting objects into perspective. In *CVPR'06*, 2006.

[10] M. Isard and A. Blake. CONDENSATION–conditional density propagation for visual tracking. *IJCV*, 29(1), 1998.

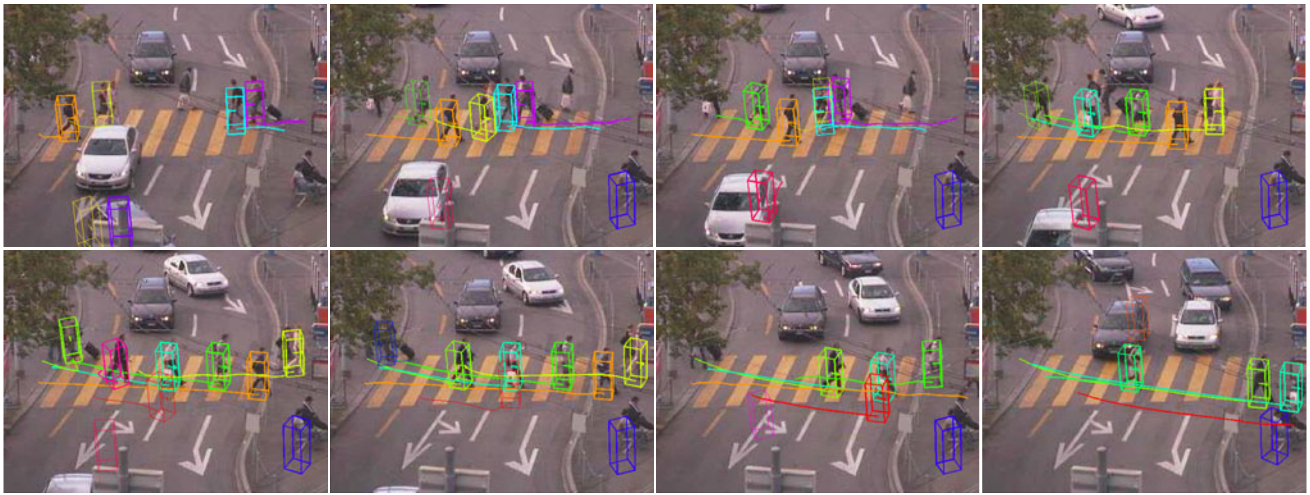

Figure 5. *Tracking results on a pedestrian crossing scenario with occlusions and background changes.*

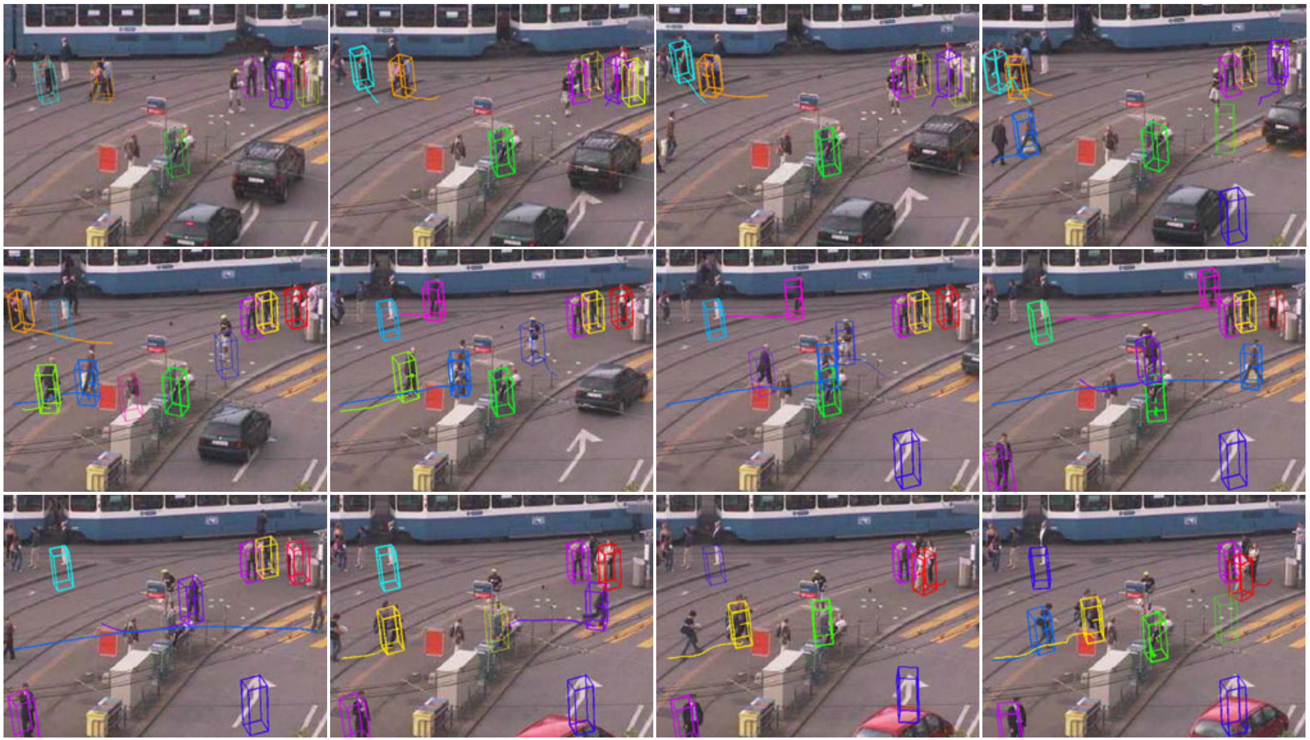

Figure 6. *Results on a challenging sequence with many static pedestrians, frequent occlusions, and large-scale background changes.*

[11] R. Kaucic, A. Perera, G. Brooksby, J. Kaufhold, and A. Hoogs. A unified framework for tracking through occlusions and across sensor gaps. In *CVPR'05*, 2005.

[12] O. Lanz. Approximate bayesian multibody tracking. *Trans. PAMI*, 28(9):1436–1449, 2006.

[13] B. Leibe, N. Cornelis, K. Cornelis, and L. Van Gool. Dynamic 3d scene analysis from a moving vehicle. In *CVPR'07*, 2007.

[14] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *CVPR'05*, 2005.

[15] A. Leonardis, A. Gupta, and R. Bajcsy. Segmentation of range images as the search for geometric parametric models. *IJCV*, 14, 1995.

[16] K. Mikolajczyk, B. Leibe, and B. Schiele. Multiple object class detection with a generative model. In *CVPR*, 2006.

[17] K. Okuma, A. Taleghani, N. de Freitas, J. Little, and D. Lowe. A boosted particle filter: Multitarget detection and tracking. In *ECCV'04*, 2004.

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

[19] K. Schindler, J. U, and H. Wang. Perspective n-view multibody structure-and-motion through model selection. In *ECCV'06*, 2006.

[20] C. Stauffer and W. Grimson. Adaptive background mixture models for realtime tracking. In *CVPR'99*, 1999.

[21] P. Viola and M. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2004.

[22] B. Wu and R. Nevatia. Tracking of multiple, partially occluded humans based on static body part detections. In *CVPR'06*, 2006.

[23] F. Yan, A. Kostin, W. Christmas, and J. Kittler. A novel data association algorithm for object tracking in clutter with application to tennis video analysis. In *CVPR'06*, 2006.