Hypergraphs for Joint Multi-View Reconstruction and Multi-Object Tracking

Martin Hofmann[1], Daniel Wolf[1,2], Gerhard Rigoll[1]
[1]Institute for Human-Machine Communication, Technische Universität München
[2]Automation & Control Institute, Technische Universität Wien
martin.hofmann@tum.de, wolf@acin.tuwien.ac.at, rigoll@tum.de

Abstract

We generalize the network flow formulation for multi-object tracking to multi-camera setups. In the past, reconstruction of multi-camera data was done as a separate extension. In this work, we present a combined maximum a posteriori (MAP) formulation, which jointly models multi-camera reconstruction as well as global temporal data association. A flow graph is constructed, which tracks objects in 3D world space. The multi-camera reconstruction can be efficiently incorporated as additional constraints on the flow graph without making the graph unnecessarily large. The final graph is efficiently solved using binary linear programming. On the PETS 2009 dataset we achieve results that significantly exceed the current state of the art.

1. Introduction

In this work, we consider the problem of tracking a variable number of objects in setups with multiple overlapping views. We follow a tracking-by-detection paradigm. Thus, given a set of object detections in each frame and from each camera, the problem of tracking becomes a data association problem. Challenges of this association problem include false positives and the fact that detections may be missing due to false negatives of the detector or due to occlusions. Most of all, however, the central challenge is the exponentially huge search space. Greedy approaches [17, 5, 12] have solved the problem based on heuristics. Global approaches [21, 16] model the data in a joint optimization framework. We continue the work of [21] which has proven to be a mathematically solid framework for global multi-object tracking. Efficient greedy [16] and globally optimal solutions [4] exist.

The main contribution in this work is an extension of the global tracking framework to setups with multiple overlapping views. Besides temporal data association, as used in the single camera case, the additional problem of data association between cameras arises in the multi-camera approach. We propose a method to jointly model multi-camera and temporal data association. This kind of generalization has recently been attempted in [14]. In their work, one tracking graph is constructed for each view and the multi-camera couplings are incorporated by an additional tracking graph for each possible camera pair. Contrary to this approach, we use only one global graph for tracking, keeping the problem as simple as possible. The nodes in this graph directly contain the feasible multi-camera couplings. To ensure that each detection is used in only one trajectory, coupling constraints on these nodes are introduced.

The tracking task is first modeled as a global maximum a posteriori problem (similar to [21]), however, the formulation is generalized to take the multi-camera couplings into account. A rearrangement into the log space leads to the constrained flow formulation, which is simple enough to be efficiently solved by a state-of-the-art general purpose binary linear programming solver.

Experimental results show that the method can successfully leverage input from multiple cameras and results outperform related methods on low, medium and dense tracking scenarios of the PETS 2009 database [8].

2. Related Work

Target tracking has been studied extensively. Local tracking approaches (where each target is tracked independently) using for example the Kalman Filter [19] have high precision and localization accuracy, but fail in multi-object scenarios where association of detections and trajectories becomes a major issue.

Joint multi-target trackers have long outperformed such independent trackers. For example, Multi-Hypothesis Tracking (MHT) [17] and Joint Probabilistic Data Association Filters (JPDAF) [9] overcome this problem by jointly optimizing trajectories, but these methods suffer from the combinatorial hypotheses space.

Another class of recent and very successful approaches define tracking as a global optimization over the complete sequence [21, 16, 10]. Here, a global posterior probability is formulated and maximized. These methods are conceptually solid and fast algorithms exist (e. g. Hungarian Algo-

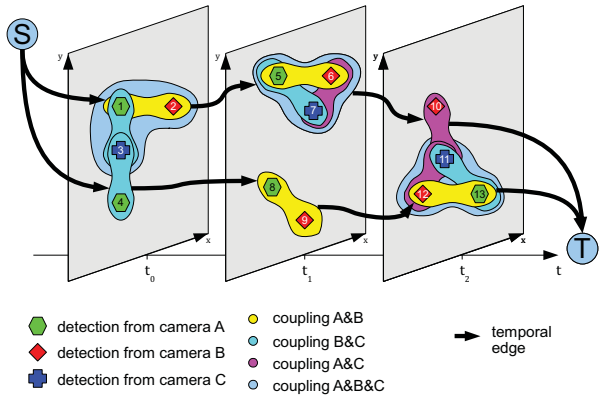


Figure 1: Example showing detections (in world-coordinates) of three cameras in three frames. Each detection $\mathbf{x}_i$ becomes a node in a hypergraph. Within each frame, hyperedges $\mathcal{R}_k$ (connecting multiple detection nodes) denote potential couplings, which correspond to the 3D reconstructions. Arrows indicate temporal tracking edges (here, only final tracking edges are shown).

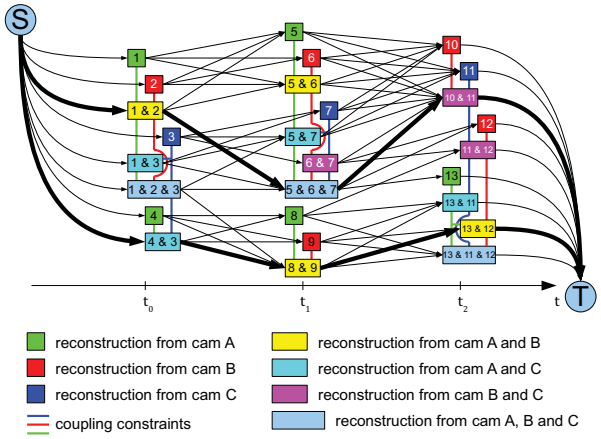


Figure 2: Example showing the network flow graph corresponding to the hypergraph in Figure 1. Each node corresponds to a 3D reconstruction $\mathcal{R}_k$. Temporal edges connect the 3D reconstructions over time. Coupling constraints ensure that an underlying 2D detection $\mathbf{x}_i$ may only be used in at most one 3D reconstruction. The final flow (bold edges) favors reconstructions which couple more 2D detections.

rithm [10], k-shortest paths [4]).

When considering multi-camera tracking, the additional problem of data association between views (i. e. *reconstruction*) arises. Thus far, reconstruction and tracking have been handled as separate stages (see [20] for a comparison). An attempt to jointly solve these two problems for multi-camera multi-target tracking has recently been presented in [14]. In this work, a separate tracking graph is constructed for each view. In addition, for each pair of cameras, an additional tracking graph is constructed, providing the coupling constraints for the involved views. And furthermore, in their approach, the size of the graph is scaled by the number of potentially tracked targets. Thus, the number of targets has to be roughly known a priori. This leads to a huge tracking graph, which relies on a specialized optimization technique. In contrast, in our work, we present a solution which only requires a single tracking graph. Multi-camera coupling constraints are incorporated into the reconstruction nodes within the tracking graph. Tracking therefore only needs to be done once in the world coordinate space.

3. Tracking

The input to the tracking stage are object detections from each frame and from each camera. For object detection we use discriminatively trained deformable part models [7]. Each 2D detection is defined by a tuple $\mathbf{x}_i = (x_i, s_i, c_i, t_i)$, where x_i and s_i are the location and size in pixel coordinates, c_i is the camera and t_i the time index. The set of all detections is $\mathcal{X} = \{\mathbf{x}_i\}$. Ideally, an object which is seen by all available cameras generates a 2D detection in each view and the corresponding projections to the common world coordinates should all come to the same location. However, due to projection errors and imprecise detections, the resulting 3D positions are unlikely to match up exactly. Furthermore, because of false positive detections as well as missing detections and occlusions, reconstruction becomes challenging. We therefore leave the coupling to the optimization stage and define the 3D reconstructions $\mathcal{R}_k$ as hyperedges on the set of object detections from each frame:

$$\mathcal{R}_k \subseteq \mathcal{X} | \, (\forall \mathbf{x}_i, \mathbf{x}_j \in \mathcal{R}_k, i \neq j : c_i \neq c_j \wedge t_i = t_j) \quad (1)$$

Only 2D detections at the same time, but in different views can be coupled in a 3D reconstruction $\mathcal{R}_k$. A 3D reconstruction $\mathcal{R}_k$ may also consist of a single 2D detection because an object does not necessarily need to be detected in more than one view. The set of all available and feasible reconstructions is $\mathcal{R} = \{\mathcal{R}_k\}$. If all theoretically possible couplings were considered, the set $\mathcal{R}$ would be huge and the whole problem intractable. Therefore, the set of reconstructions is reduced to reconstructions $\mathcal{R}_k$ which have a prior reconstruction probability $P_{rec}(\mathcal{R}_k) > 0$ as defined in Section 5.1.

3.1. MAP Formulation

We define a single trajectory hypothesis $\mathcal{T}_u$ as an ordered list of 3D reconstructions $\mathcal{T}_u = \{\mathcal{R}_{u_1}, \mathcal{R}_{u_2}, \ldots, \mathcal{R}_{u_{n_u}}\}$.

The complete association hypothesis $\mathcal{T}$ is then defined as a set of trajectory hypotheses, thus $\mathcal{T} = \{\mathcal{T}_u\}$. The joint reconstruction and tracking is achieved by maximizing the posterior probability of the association hypothesis $\mathcal{T}$, given the set of reconstructions $\mathcal{R}$:

$$\begin{aligned}\mathcal{T}^* &= \arg\max_{\mathcal{T}} P(\mathcal{T}|\mathcal{R}) \\ &= \arg\max_{\mathcal{T}} P(\mathcal{R}|\mathcal{T})P(\mathcal{T}) \end{aligned} \tag{2}$$

The individual likelihood probabilities $P(\mathcal{R}_k|\mathcal{T})$ are conditionally dependent because two 3D reconstructions $\mathcal{R}_k$ and $\mathcal{R}_l$ may both contain one or more of the same 2D detections $\mathbf{x}_i$. By construction it has to be ensured that each 2D detection $\mathbf{x}_i$ is used at most for one 3D reconstruction which is part of a trajectory. This can be efficiently captured in the *coupling constraint*:

$$\mathcal{R}_k \cap \mathcal{R}_l = \emptyset,\ \forall k \neq l,\ \forall \mathcal{R}_k, \mathcal{R}_l \in \mathcal{T} \tag{3}$$

With this constraint, the likelihood terms become independent and Equation (2) can be written as:

$$\mathcal{T}^* = \arg\max_{\mathcal{T}} \prod_{\mathcal{R}_k \in \mathcal{R}} P(\mathcal{R}_k|\mathcal{T})P(\mathcal{T}) \tag{4}$$

With the additional *non-overlap constraint*, which ensures that a reconstruction $\mathcal{R}_k$ can only be part of at most one trajectory, i. e.

$$\mathcal{T}_u \cap \mathcal{T}_v = \emptyset,\ \forall u \neq v \tag{5}$$

and with the assumption that the motions of all objects do not depend on each other, the prior for trajectories $P(\mathcal{T})$ can be further factorized. Thus, the final MAP formulation can be written as follows:

$$\begin{aligned}\mathcal{T}^* = \arg\max_{\mathcal{T}} & \prod_{\mathcal{R}_k \in \mathcal{R}} P(\mathcal{R}_k|\mathcal{T}) \prod_{\mathcal{T}_u \in \mathcal{T}} P(\mathcal{T}_u) \\ \text{s.t. } & \mathcal{R}_k \cap \mathcal{R}_l = \emptyset,\ \forall k \neq l,\ \forall \mathcal{R}_k, \mathcal{R}_l \in \mathcal{T} \\ & \mathcal{T}_u \cap \mathcal{T}_v = \emptyset,\ \forall u \neq v \end{aligned} \tag{6}$$

Here, $P(\mathcal{R}_k|\mathcal{T}) = P_{det}(\mathcal{R}_k|\mathcal{T}) \cdot P_{rec}(\mathcal{R}_k|\mathcal{T})$ is the likelihood of the 3D reconstruction $\mathcal{R}_k$, which can be decomposed into a detection likelihood term, as well as into a reconstruction likelihood term.

The detection likelihood takes the probability of each underlying 2D detection being a true detection or a false positive, as well as the likelihood of missed detections into account:

$$P_{det}(\mathcal{R}_k|\mathcal{T}) = \begin{cases} (1-\beta)^{|\mathcal{R}_k|} \cdot \gamma^{n(\mathcal{R}_k)-|\mathcal{R}_k|}, & \mathcal{R}_k \in \mathcal{T} \\ \beta^{|\mathcal{R}_k|} \cdot (1-\gamma)^{n(\mathcal{R}_k)-|\mathcal{R}_k|}, & \text{else} \end{cases} \tag{7}$$

where β and γ are the false positive and false negative rates of the detector and $n(\mathcal{R}_k)$ is the number of cameras that *should* generate a detection for the 3D reconstruction $\mathcal{R}_k$.

The quality of the reconstruction is measured by the reconstruction likelihood:

$$P_{rec}(\mathcal{R}_k|\mathcal{T}) = \begin{cases} P_{rec}(\mathcal{R}_k), & \mathcal{R}_k \in \mathcal{T} \\ (1 - P_{rec}(\mathcal{R}_k)), & \text{else} \end{cases} \tag{8}$$

Here, $P_{rec}(\mathcal{R}_k)$ is the a priori reconstruction probability. It is defined in Section 5.1.

As it is typically done in tracking approaches, the a priori likelihood of a single trajectory hypothesis $P(\mathcal{T}_u)$ is given by a Markov chain:

$$\begin{aligned} P(\mathcal{T}_u) &= P(\{\mathcal{R}_{u_0}, \mathcal{R}_{u_1}, \ldots, \mathcal{R}_{u_{n_u}}\}) \\ &= P_{en}(\mathcal{R}_{u_0}) P_{link}(\mathcal{R}_{u_1}|\mathcal{R}_{u_0}) \ldots P_{ex}(\mathcal{R}_{u_{n_u}}) \end{aligned} \tag{9}$$

$P_{en}(\mathcal{R}_{u_i})$ and $P_{ex}(\mathcal{R}_{u_i})$ define the probability of a trajectory to start and end at reconstruction $\mathcal{R}_{u_i}$, respectively. $P_{link}(\mathcal{R}_{u_j}|\mathcal{R}_{u_i})$ defines the transition probability from reconstruction $\mathcal{R}_{u_i}$ to $\mathcal{R}_{u_j}$. See Section 5.2 and 5.3.

4. Mapping to a constrained Min-Cost Flow Graph

The final MAP formulation of Equation (6), which corresponds to a hypergraph like the one in Figure 1, can be efficiently reformulated into a constrained min-cost flow graph (as the one in Figure 2). Using this min-cost flow formulation, the tracking problem can be efficiently solved using binary linear programming algorithms.

The basic idea of mapping the MAP formulation into a cost-flow network is that the non-overlap constraint, given in Equation (5), corresponds to edge-disjoint paths in a directed graph.

The min-cost flow graph is built of the hyperedges $\mathcal{R}_k$ (corresponding to 3D reconstructions), as well as the temporal edges $\mathcal{E}_{k,l} = \{\mathcal{R}_k, \mathcal{R}_l\}$, which connect 3D reconstructions between frames. In addition, each hyperedge $\mathcal{R}_k$ can connect to a source node S (via a source edge) and to a sink node T (via a sink edge) to mark the beginning and the end of a trajectory, respectively. Each reconstruction hyperedge, each temporal edge as well as each source and sink edge can carry a certain amount of flow f generating a cost per flow unit c. Thus, to be precise, the reconstruction hyperedges $\mathcal{R}_k$ have flow f_k with associated cost C_k. Each temporal edge $\mathcal{E}_{k,l}$ has a flow of $f_{k,l}$ with a cost of $C_{k,l}$ and analogously, the source and sink edges have flow of $f_{en,k}$ and $f_{ex,k}$, with a cost of $C_{en,k}$ and $C_{ex,k}$, respectively.

Each flow path through the graph corresponds to an object trajectory and the total amount of flow from S to T represents the number of tracked trajectories. As we impose the constraint that each trajectory can only belong to one object and vice versa, the flow f through an edge can be either 0 or 1. A flow of $f = 1$ implies that the corresponding edge is part of the trajectory, a flow of $f = 0$ means that the edge (or hyperedge) is not used.

To account for the flow conservation constraint (defined in Equation (5)) the sum of the outgoing flows of a hyperedge $\mathcal{R}_k$ equals the sum of the incoming flows:

$$f_{en,k} + \sum_l f_{l,k} = f_k = f_{ex,k} + \sum_l f_{k,l}, \quad \forall k \tag{10}$$

Recall that in the MAP formulation the coupling constraint of Equation (3) was introduced to ensure that every 2D detection $\mathbf{x}_i$ can only be used for one 3D reconstruction $\mathcal{R}_k$. This coupling constraint is translated to the flow graph representation in the following way: For all 3D reconstructions $\mathcal{R}_k$ and $\mathcal{R}_l$, which have at least one 2D detection in common (i. e. $\mathcal{R}_k \cap \mathcal{R}_l \neq \emptyset$), the sum of the corresponding flows f_k, f_l must be either 0 or 1. This ensures that at most one of these 3D reconstructions can be used. Formally, this constraint translates to a flow constraint as follows:

$$\sum_{k \in Q_i} f_k \leq 1, \quad \forall i,\ Q_i = \{k | \mathbf{x}_i \in \mathcal{R}_k\} \tag{11}$$

The problem of maximizing the probability of the assignment hypothesis $\mathcal{T}$ can now be reformulated as the problem of finding the best flow through the flow graph. The corresponding flow costs emerge after taking the negative logarithm on the MAP formulation:

$$\begin{aligned}
\mathcal{T}^* &= \arg\min_{\mathcal{T}} \sum_{\mathcal{R}_k \in \mathcal{T}_u} -\log P(\mathcal{R}_k|\mathcal{T}) + \sum_{\mathcal{T}_u \in \mathcal{T}} -\log P(\mathcal{T}_u) \\
&= \arg\min_{\mathcal{T}} \sum_{\mathcal{R}_k \in \mathcal{T}_u} -\log P(\mathcal{R}_k|\mathcal{T}) + \sum_{\mathcal{T}_u \in \mathcal{T}} \big(-\log P_{en}(\mathcal{R}_{u_0}) \\
&\qquad + \sum_j -\log P_{link}(\mathcal{R}_{u_{j+1}}|\mathcal{R}_{u_j}) - \log P_{ex}(\mathcal{R}_{u_{l_u}})\big) \\
&= \arg\min_{\mathcal{T}} \sum_k C_k f_k + \sum_k C_{en,k} f_{en,k} \\
&\qquad + \sum_{k,l} C_{k,l} f_{k,l} + \sum_k C_{ex,k} f_{ex,k}
\end{aligned} \tag{12}$$

subject to Equation (10) and (11). With this, the costs C naturally emerge from the MAP formulation:

$$C_{en,k} = -\log P_{en}(\mathcal{R}_k) \tag{13}$$

$$C_{k,l} = -\log P_{link}(\mathcal{R}_l|\mathcal{R}_k) \tag{14}$$

$$C_{ex,k} = -\log P_{ex}(\mathcal{R}_k) \tag{15}$$

$$\begin{aligned}
C_k = |\mathcal{R}_k| \log \frac{\beta}{1-\beta} + (n(\mathcal{R}_k) - |\mathcal{R}_k|) \log \frac{1-\gamma}{\gamma} \\
+ \log \frac{1 - P_{rec}(\mathcal{R}_k)}{P_{rec}(\mathcal{R}_k)}
\end{aligned} \tag{16}$$

The first term of C_k can be seen as a bonus for the detection, since it is negative for $\beta < 0.5$. The second term is a penalty for missed detections. The last term models the reconstruction cost, which is negative (thus a bonus) for small reconstruction errors ($P_{rec}(\mathcal{R}_k) > 0.5$).

Thus, the definition of C_k favors observations reconstructed from more than one view.

With Equation (12), our tracking and reconstruction problem is completely defined as a binary integer programming problem (BIP). BIPs as the presented one are solvable using cutting-plane methods, branch and cut or branch and price methods. In our practical implementation we use the MATLAB bintprog solver (which uses a branch-and-bound algorithm) and the faster CPLEX binary integer programming solver.

5. Modeling of Probabilities

The presented tracking model requires a definition of the reconstruction probabilities P_{rec}, the transition probabilities P_{link}, as well as the enter and exit probabilities P_{en} and P_{ex}. These are precisely defined in the following.

5.1. Reconstruction Probability P_{rec}

The prior reconstruction probability $P_{rec}(\mathcal{R}_k)$ measures the a priori quality of a 3D reconstruction $\mathcal{R}_k$ based on the mutual and absolute positions of the 2D detections $\mathbf{x}_i \in \mathcal{R}_k$ contained within a 3D reconstruction set. Ideally, all 2D detections within a 3D reconstruction set should map to the same position in world coordinates. Thus, high deviations are penalized. However, the quality of a 3D reconstruction also largely depends on the localization error ε_{det} of the object detector, as well as on the camera calibration error ε_{cal}.

First, the reconstruction error ε_k is defined as the root mean square deviation from the mean position of the detections within the set $\mathcal{R}_k$:

$$\varepsilon_k = \varepsilon(\mathcal{R}_k) = \sqrt{\frac{1}{|\mathcal{R}_k|} \sum_{\mathbf{x} \in \mathcal{R}_k} |\Phi^c(\mathbf{x}) - \chi_k|^2} \tag{17}$$

$$\chi_k = \chi(\mathcal{R}_k) = \frac{1}{|\mathcal{R}_k|} \sum_{\mathbf{x} \in \mathcal{R}_k} \Phi^c(\mathbf{x}) \tag{18}$$

Here, $\Phi^c(\mathbf{x})$ is the transformation function from image to world coordinates for camera c and $\chi(\mathcal{R}_k)$ is the reconstructed average 3D position of the coupled 2D detections in $\mathcal{R}_k$.

The prior reconstruction probability $P_{rec}(\mathcal{R}_k)$ maps the reconstruction error ε_k to a probability. Like in [14], we use a decreasing function $\mathcal{F}$ for the mapping:

$$\mathcal{F}(d, d_{min}, d_{max}) = \frac{1}{2}\text{erfc}\left(4\frac{d - d_{min}}{d_{max} - d_{min}} - 2\right) \tag{19}$$

With this mapping function, the prior reconstruction probability is defined as:

$$P_{rec}(\mathcal{R}_k) = \begin{cases} \mathcal{F}(\varepsilon(\mathcal{R}_k), 0, \varepsilon_{max}(\mathcal{R}_k)) & |\mathcal{R}_k| > 1 \\ 0.5 & \text{else} \end{cases} \tag{20}$$

For 3D reconstructions which originate from a single 2D detection no reconstruction error exists. In this case $P_{rec}(\mathcal{R}_k) = 0.5$, such that in Equation (16) no bonus is given to reconstructions with a single detection. As can be seen, the reconstruction probability decreases as the 3D reconstruction error increases. The maximal allowed reconstruction error is denoted as ε_{max}. Thus, for $\varepsilon > \varepsilon_{max} \Leftrightarrow P_{rec} = 0$.

As stated above, the reconstruction probability also largely depends on the detector inaccuracies (error in the 2D image plane) as well as the error of the camera calibration (error in the 3D world coordinate space). To take these two influences into account, the maximum allowed reconstruction error $\varepsilon_{max}(\mathcal{R}_k)$ is modeled as a function of the set of coupled 2D detections.

$$\varepsilon_{max}(\mathcal{R}_k) = \varepsilon_{det} \cdot \sum_{\mathbf{x} \in \mathcal{R}_k} \|\Theta^c(\mathbf{x})\| + \varepsilon_{cal} \tag{21}$$

Here, ε_{cal} models the calibration error in world coordinates. It depends on the quality of the camera calibration and is set to a constant. The detection error ε_{det} is an error in image coordinates, which is due to inaccurate 2D object detections. Because of the perspective distortion of the camera calibration, the 2D detection error ε_{det} has different influence on the world coordinates depending on the object location and on the camera position.

To model this effect, we calculate the sensitivity of the 2D→3D projection function $\Phi^c(\mathbf{x})$ at each image position $\mathbf{x}$. This sensitivity function corresponds to the *Jacobian matrix* of $\Phi^c(x)$:

$$\frac{\partial}{\partial \mathbf{x}} \Phi^c(\mathbf{x}) = J^c(\mathbf{x}) = \begin{pmatrix} \frac{\partial \Phi^c_{x_w}}{\partial x} & \frac{\partial \Phi^c_{x_w}}{\partial y} \\ \frac{\partial \Phi^c_{y_w}}{\partial x} & \frac{\partial \Phi^c_{y_w}}{\partial y} \end{pmatrix} \tag{22}$$

As we are just interested in the magnitude of the largest change and not in the direction of the change in x we can define the sensitivity function $\Theta^c(x)$ as the vector of lengths of the gradients of the x_w and y_w component of Φ^c:

$$\Theta^c(\mathbf{x}) = \begin{pmatrix} \|\nabla_{x_w} \Phi^c(\mathbf{x})\| \\ \|\nabla_{y_w} \Phi^c(\mathbf{x})\| \end{pmatrix} \tag{23}$$

In Equation (21), the constant detection error ε_{det} is weighted by the sum of the absolute sensitivity functions $\Theta^c(\mathbf{x})$ of all cameras that contribute to $\mathcal{R}_k$.

5.2. Entrance and Exit Probabilities P_{en} and P_{ex}

The entrance and exit probabilities of an observation define the probability of a trajectory to start and end at this observation, respectively. P_{en} and P_{ex} are modeled to account for the following considerations: (1) Observations which are close to the boundary of the tracking area are likely to enter or exit the scene. (2) Objects, which have about the size of the minimum detection size of the underlying object detector are likely to mark the beginning or end of a trajectory. (3) All observations in the first frame have a high entrance probability and all observations in the last frame have a high exit probability.

5.3. Transition Probabilities P_{link}

The transition probability $P_{link}(\mathcal{R}_l|\mathcal{R}_k)$ defines the probability of two 3D reconstructions $\mathcal{R}_k$ and $\mathcal{R}_l$ to be part of a single object trajectory at different (subsequent) time steps. In our work, this probability only depends on the spatial and temporal distance between the two observations (i. e. no appearance or motion direction):

$$P_{link}(\mathcal{R}_l|\mathcal{R}_k) = P(\chi_l|\chi_k, \Delta\tau) P(\Delta\tau) \tag{24}$$

The spatial term is modeled by the distance probability function (using Equation (19)):

$$P(\chi_l|\chi_k, \Delta\tau) = \mathcal{F}\left(\|\chi_k - \chi_l\|, 0, \frac{v_{max}}{f}\Delta\tau\right) \tag{25}$$

where v_{max} is the maximum 3D velocity of a person, f is the video frame rate and $\Delta\tau = \tau_l - \tau_k$ is the frame gap. Similar to [21] we describe the temporal term with an exponential model:

$$P(\Delta\tau) = \begin{cases} \gamma^{n(\Delta\tau - 1)} & , 1 \leq \Delta\tau \leq \Delta\tau_{max} \\ 0 & \text{else} \end{cases} \tag{26}$$

where $\Delta\tau_{max}$ is the maximal allowed frame difference between two observations, γ is the false negative rate of the detector and n is the (average) number of cameras that *should* have seen the object in the frame gap.

5.4. Tracking Post-Processing

After the BIP optimization, trajectories are defined by tracking back along the active edges ($f = 1$). In some frames, an object might not have been detected in all views (e. g. due to occlusion). An estimated 2D bounding box can be recovered (for each missing view) from the average 3D position χ_k (see Equation (18)). Due to detection and calibration inaccuracies the resulting trajectories might be jagged. They can also have gaps of several frames, if an object was occluded in all views at the same time. Therefore, the 3D coordinates of all trajectories as well as the 2D bounding box coordinates in each view are smoothed using a Savitzky-Golay filter [18]. Finally, to fill the gaps, we perform linear interpolation in 2D and in 3D to get coordinates for each object in each frame and view.

6. Evaluation

We evaluate our multi-camera multi-person tracking system on the publicly available PETS 2009 dataset [8]. It

contains different scenarios with three levels of difficulty with low (S2.L1), medium (S2.L2) and high person densities (S2.L3). PETS 2009 is a very challenging dataset for multiple person tracking as there are many inter-object occlusions, especially in S2.L2 and S2.L3. The scene contains a static object right in the middle (a light pole with a big sign) occluding persons walking behind it. The frame rate of the videos is only 7 frames per second, so persons can move quite far between two consecutive frames making precise tracking even more challenging. While many previously presented approaches are merely evaluated on the low and medium density scenarios, we are also able to show results for the high density scenes.

6.1. Performance Metrics for Multiple Object Tracking

We use the widespread CLEAR measures introduced in [13] called Multiple Object Tracking Accuracy (MOTA) and Multiple Object Tracking Precision (MOTP). In addition, we use the metrics that were presented in [15]. These are Identity Switches (IDS), Track Fragments (FM), Mostly Tracked (MT), Partly Tracked (PT) and Mostly Lost (ML).

We use the ground truth used in [1]. Assignment of tracking output to ground truth is done using the Hungarian algorithm with an assignment cut-off at 1 meter. MOTP is normalized to this cut-off threshold.

We found MOTA, FM and IDS to be the most meaningful of the listed measures as they best measure the quality of a stable tracking of identities over a whole scenario. MOTP, by contrast, merely measures the mean distance of the trajectories to the closest ground truth and therefore largely depends on the annotation quality.

6.2. Experimental Settings

The tracking algorithm can in principle be used with arbitrarily many views. We show results for using one, two and three cameras for each of the three tracking scenarios.

The method uses several parameters to model the observed scenario, including the quality of the detector and the camera calibration. The parameters are intuitive and we use the following default settings for all scenarios.

Default settings The maximal walking speed of a person is limited to $v_{max} = 5\,\text{m/s}$, such that tracking a running person is possible.

All observations closer than $d_{b,max} = 1\,\text{m}$ to the boundary of the observed scene are considered to be "close to the boundary" as well as all observations with a maximum detection height being smaller than $\alpha = 2$ times the minimum detectable height. For all enter and exit probabilities, we set the maximum value to $P_{en,max} = P_{ex,max} = 0.1$.

We expect the detector to have an average bounding box inaccuracy of $\varepsilon_{det} = 4\,\text{px}$ and set the expected calibration error to $\varepsilon_{cal} = 0.5\,\text{m}$.

(a) S2.L1 (b) S2.L2 (c) S2.L3

Figure 3: Tracking results on PETS 2009 for three cameras and ground plane; (a) low density (S2.L1, camera 1+5+7), (b) medium density (S2.L2, camera 1+2+3), (c) high density (S2.L3, camera 1+2+4); solid box: detected in this camera; dotted box: reconstructed from other camera; dashed box: interpolated (detected in no camera).

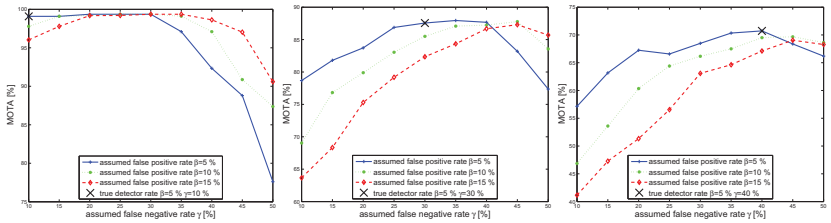
The false positive rate β, and the false negative rate γ are estimated by evaluating the detector output against the available ground truth for each scenario. Using this evaluation, we found for all scenarios an almost constant false positive rate of $\beta = 0.05$. The false negative rate, however, significantly deviates for each scenario. For S2.L1 $\gamma = 0.1$, for S2.L2 $\gamma = 0.3$ and for S2.L3 $\gamma = 0.4$. These false negative rates are reasonable since in more crowded scenes, the used detector will by far not be able to find enough detections due to mutual occlusions.

The maximum frame gap is set to $\Delta\tau_{max} = 9$. A higher frame gap leads to more interconnections in the graph and slows down calculation, a significantly lower frame gap reduces performance.

Influence of parameters Most of the used parameters (such as maximum walking speed, distance to tracking boundary, etc.) can be set by intuition and have little impact on tracking performance. The parameters which show the most significant influence are the false positive rate $P_{fp} = \beta$ and the false negative rate $P_{fn} = \gamma$, which highly depend on the detector quality. We keep the detector setting constant (i. e. keep the same object detections) and tune the parameters β and γ. The tracking results for low, medium

Sequence	Method	Camera IDs	MOTA [%]	MOTP [%]	MT [%]	PT [%]	ML [%]	FM	IDS
PETS S2.L1	Andriyenko et al. [1]	1	88.3	75.7	86.96	4.35	8.70	-	-
	Andriyenko et al. [2]	1	95.9	78.7	100.0	0.0	0.0	8	10
	Berclaz et al. [3]	1+3+5+6+8	82	56	-	-	-	-	-
	Breitenstein et al. [5]	1	75	60	-	-	-	-	-
	Hofmann et al. [11]	1	97.8	75.3	100	0	0	8	8
	Leal-Taixé et al. [14]	1+5	76.0	60	-	-	-	-	-
	Leal-Taixé et al. [14]	1+5+6	71.4	53.4	-	-	-	-	-
	our (1 camera)	1	98.0	82.8	**100.0**	**0.0**	**0.0**	11	10
	our (2 cameras)	1+5	**99.4**	82.9	**100.0**	**0.0**	**0.0**	**1**	**1**
	our (3 cameras)	1+5+7	**99.4**	**83.0**	**100.0**	**0.0**	**0.0**	**1**	2
PETS S2.L2	Andriyenko [1]	1	60.2	60.5	33.33	56	10.67	-	-
	Hofmann et al. [11]	1	57.1	56.4	39.5	42.1	18.4	**59**	**67**
	our (1 camera)	1	75.8	72.1	65.1	34.9	**0.0**	252	234
	our (2 cameras)	1+2	**87.6**	73.5	**86.0**	14.0	**0.0**	128	111
	our (3 cameras)	1+2+3	79.7	**74.2**	69.8	27.9	2.3	129	132
PETS S2.L3	Hofmann et al. [11]	1	41.5	65.0	34.1	34.1	31.8	**67**	**46**
	our (1 camera)	1	62.8	70.5	**54.5**	34.1	**11.4**	217	225
	our (2 cameras)	1+2	**68.5**	72.3	**54.5**	25.0	20.5	149	156
	our (3 cameras)	1+2+4	65.4	**73.9**	40.9	34.1	25.0	88	116

Table 1: Quantitative results on the PETS 2009 database. Results are compared to "Continuous Energy Minimization" [1, 2], "Probabilistic Occupancy Maps" [3], "Particle filter based tracking-by-detection" [5], "Hierarchical data association" [11] and "Branch-and-price global optimization" [14]. The results of [3, 5] are taken from Figure 3 in [6].



(a) S2.L1 (low density) (b) S2.L2 (medium density) (c) S2.L3 (high density)

Figure 4: Sensitivity of MOTA to the assumed false positive rate $P_{fp} = \beta$ and the assumed false negative rate $P_{fn} = \gamma$ for low, medium and high density scenes. For the actual (true) false positive and false negative rates of the detector (obtained using ground truth), best results are in fact achieved.

scenario	S2.L1	S2.L2	S2.L3
length	120	62	35
1 camera	**19**	**26**	**2**
2 cameras	**41**	101	36
3 cameras	**80**	974	944

Table 2: Runtime (in seconds) of the CPLEX solver on an i5 2.6 Ghz, 12 GB RAM. Bold indicates faster than scenario length.

and high density scenarios with these "assumed" false positive and false negative rates are shown in Figure 4. It can be seen that for low density scenarios, these parameters have little influence on the system performance for a wide range of settings. For medium and high density scenarios, parameter tuning has an impact on performance. It can be seen that the β and γ values, found using the ground truth (as described above), are in fact (almost) the same as the ones found using parameter tuning. Thus, if the false positive and false negative rates can be estimated correctly on a given dataset, the tracker can get the most out of the erroneous detections.

6.3. Overall Results

Qualitative tracking results for S2.L1, S2.L2 and S2.L3 using three cameras are shown in Figure 3. Quantitative results using one, two and three cameras are given in Table 1. Since the goal of the proposed method is to leverage multiple cameras, it is no surprise that using two or three cameras greatly outperforms the case when only one camera is used. Especially since no appearance and explicit long term occlusion terms are used in our approach.

It can be seen that, with the presented multi-camera tracker, the low density tracking scene S2.L1 achieves excellent results with a MOTA of 99.4%. Using more than one camera, only one ID switch and one fragment remain. These are due to the fact that an object is not annotated at the boundary, where the tracker however still successfully tracks a person. Also, the MOTP of 83.0% in the three camera version indicates a very reliable tracking precision. Using more cameras leads to slightly higher tracking precision (MOTP) since localization information from multiple sources can be used.

Regarding the more challenging scenarios, and using two

cameras, we still see very good tracking accuracy with a MOTA of 87.6% for S2.L2 (medium density) and 68.5% for S2.L3 (high density). However, MOTA goes down when using three cameras instead of two cameras. This phenomenon (also observed in [14]) seems to persist mainly in more crowded scenarios, where calibration errors of multiple sources add up and lead to assignment errors. On the other hand, like in S2.L1, using more cameras slightly increases tracking precision (MOTP) due to more available localization data.

A relatively high number of fragments and ID switches is to be observed in S2.L2 and S2.L3. Here, the high false negative rate of the detector stage (about 30%-40%) cannot be fully recovered in the tracking stage. In these dense crowds, many detections have to be "hallucinated", which leads to id switches. Appearance and motion information as well as long term occlusion modeling could potentially leverage the problem, as shown in [11], where (see Table 1) a significantly lower rate of ID switches and fragments can be obtained.

Most other approaches have mainly been evaluated on the simpler S2.L1 scenario. Only a few other approaches report quantitative results on the more difficult medium and high-density scenarios S2.L2 and S2.L3. In Table 1 it can be seen that the proposed method performs favorably compared to other methods in almost all measures.

Runtime of the CPLEX solver on an i5 CPU, 2.6 GHz, 12 GB RAM can be seen in Table 2. Thus, on the easy S2.L1, the optimization runs in less time than the scenario length.

7. Conclusion and Outlook

In this paper we have contributed to the field of tracking by global data association. We extend the well established global tracking method to multi-camera setups, which so far has only been attempted in a few works. In contrast to these works, our formulation only requires a single tracking graph and no prior knowledge about the number of tracking targets is needed. The two steps of 3D reconstruction and tracking, which previously have been treated in separate steps, are solved in a joint framework. The multi-camera detection data, which can be represented as a hypergraph, can be efficiently mapped to a constrained cost flow graph, which can be solved using standard optimization techniques.

Future work should address the object detection stage (which is not suited for high density scenes with frequent occlusions). Currently, in our experimental implementation, only three cameras are used and future work can leverage more cameras for better performance. Furthermore, performance gains can be expected when appearance and motion information is incorporated into the tracking formulation.

Acknowledgement This work has been partially funded by the European Commission under FP7-IST-288146 HOBBIT.

References

[1] A. Andriyenko, S. Roth, and K. Schindler. An analytical formulation of global occlusion reasoning for multi-target tracking. In *ICCV Workshops*, pages 1839–1846. IEEE, 2011. 6, 7

[2] A. Andriyenko, K. Schindler, and S. Roth. Discrete-continuous optimization for multi-target tracking. In *CVPR*, 2012. 7

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

[4] J. Berclaz, E. Turetken, F. Fleuret, and P. Fua. Multiple object tracking using k-shortest paths optimization. *TPAMI*, 2011. 1, 2

[5] M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. *TPAMI*, 33(9):1820 –1833, sept. 2011. 1, 7

[6] A. Ellis and J. Ferryman. Pets2010 and pets2009 evaluation of results using individual ground truthed single views. In *AVSS*, pages 135 – 142, 29 2010-sept. 1 2010. 7

[7] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *TPAMI*, 32(9):1627–1645, 2010. 2

[8] J. Ferryman and A. Shahrokni. Pets2009: Dataset and challenge. In *Winter-PETS*, pages 1–6, 2009. 1, 5

[9] T. Fortmann, Y. Bar-Shalom, and M. Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *Oceanic Engineering, IEEE Journal of*, 8(3):173 – 184, jul 1983. 1

[10] J. F. Henriques, R. Caseiro, and J. Batista. Globally optimal solution to multi-object tracking with merged measurements. In *ICCV*, pages 2470–2477. IEEE, 2011. 1, 2

[11] M. Hofmann, M. Haag, and G. Rigoll. Unified hierarchical multi-object tracking using global data association. *PETS*, 2013. 7, 8

[12] M. Hofmann, M. Kaiser, H. Aliakbarpour, and G. Rigoll. Fusion of multi-modal sensors in a voxel occupancy grid for tracking and behaviour analysis. *12th International Workshop on Image Analysis for Multimedia Interactive Services, Delft, The Netherlands*, 2011. 1

[13] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, and J. Zhang. Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. *TPAMI*, 31(2):319–336, Feb. 2009. 6

[14] L. Leal-Taixe, G. Pons-Moll, and B. Rosenhahn. Branch-and-price global optimization for multi-view multi-object tracking. In *CVPR*, June 2012. 1, 2, 4, 7, 8

[15] Y. Li, C. Huang, and R. Nevatia. Learning to associate: Hybrid-boosted multi-target tracker for crowded scene. In *CVPR*, pages 2953–2960. IEEE, 2009. 6

[16] H. Pirsiavash, D. Ramanan, and C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR*, pages 1201 –1208, june 2011. 1

[17] D. Reid. An algorithm for tracking multiple targets. *Automatic Control, IEEE Transactions on*, 24(6):843 – 854, dec 1979. 1

[18] A. Savitzky and M. J. E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36:1627–1639, 1964. 5

[19] G. Welch and G. Bishop. An introduction to the kalman filter. Technical report, Chapel Hill, NC, USA, 1995. 1

[20] Z. Wu, N. I. Hristov, T. H. Kunz, and M. Betke. Tracking-reconstruction or reconstruction-tracking? comparison of two multiple hypothesis tracking approaches to interpret 3d object motion from several camera views. In *WACV*, 2009. 2

[21] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *CVPR*. IEEE Computer Society, 2008. 1, 5