Global Data Association for Multi-Object Tracking Using Network Flows

Li Zhang, Yuan Li and Ramakant Nevatia
University of Southern California
Institute of Robotics and Intelligent Systems
{li.zhang|yli8|nevatia}@usc.edu

Abstract

We propose a network flow based optimization method for data association needed for multiple object tracking. The maximum-a-posteriori (MAP) data association problem is mapped into a cost-flow network with a non-overlap constraint on trajectories. The optimal data association is found by a min-cost flow algorithm in the network. The network is augmented to include an Explicit Occlusion Model(EOM) to track with long-term inter-object occlusions. A solution to the EOM-based network is found by an iterative approach built upon the original algorithm. Initialization and termination of trajectories and potential false observations are modeled by the formulation intrinsically. The method is efficient and does not require hypotheses pruning. Performance is compared with previous results on two public pedestrian datasets to show its improvement.

1. Introduction

Robust detection and tracking of objects are important for many computer vision tasks. We consider an approach where object detection results are given in each frame as input and the task is to associate the detections to find object trajectories. Not all objects can be expected to be detected in each frame, false detections may be present and some objects may be occluded by others; these factors make data association a difficult task.

Some methods, e.g. [1, 2], attempt to resolve ambiguities in each frame. Others, e.g. [3, 4, 5, 6, 7, 8, 9, 10] use more global information. However, the search space of those alternatives grows exponentially with the number of frames which requires severely limiting the search window and pruning of hypotheses. They also typically assume that all detections are correct which is not always accurate.

We propose an efficient global data association approach that can find optimal solutions for much longer sequences (windows) than has been possible from earlier approaches.

Figure 1. Detection input and tracking result: our method can remove false alarms, recover trajectories and infer events such as missed detections and occlusions.

In our approach, data association is defined as a MAP estimation problem given a set of object detection results as input observations. Non-overlapping trajectory hypotheses are modeled as disjoint flow paths in a cost-flow network; observation likelihood and transition probabilities are modeled as flow costs. Global optimal trajectory association is found by a min-cost flow algorithm. To track through long-term occlusions, an Explicit Occlusion Model (EOM) is constructed, by adding occlusion nodes and constraints to the network (we only consider inter-object occlusions). A minimal cost flow in the EOM-based network is solved by an iterative approach built upon the original min-cost flow algorithm. Trajectory initialization, termination and inference of object occlusions are inherent in the method, and hence can be inferred from the solution. An example of inference of occlusions and missed detections from the tracking result is shown in Figure 1.

The rest of the paper is organized as follows. Related work is discussed in Section 2. The MAP formulation and the global optimal solution are described in Section 3. The Explicit Occlusion Model and an iterative solution for it are introduced in Section 4. Implementation details are given in Section 5. Experimental results are shown in Section 6. Conclusions are given in Section 7.

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

2. Related work

To track multiple objects, one approach is to make data association decisions frame-by-frame (or in a small time window) as in [1, 2]. While such methods have shown very good performance, considering more frames before making association decisions should generally help better overcome ambiguities caused by longer-term occlusions and false or missed detections.

Many global approaches that use more information have been explored to overcome errors of detections. One strategy is to optimize one trajectory at a time through the entire sequence; this has been used in Dynamic Programming based methods, such as [5, 6]. Greedy strategies are then used to combine the trajectories and handle potential conflicts. It is difficult for these methods to model occlusions because trajectories are optimized separately. Another approach is to optimize multiple trajectories simultaneously; multi-Hypothesis Tracking (MHT) [3] and Joint Probabilistic Data Association Filters (JPDAF)[4] are two representative examples. Also in [10], detection and estimation of trajectory hypotheses are coupled by Quadratic Boolean Programming. As the hypotheses search space is combinatorial, such methods can only optimize over a limited time window, and hypotheses must still be pruned. Sampling methods such as MCMC[9] have also been employed to find approximate solutions. Occlusions are usually modeled as merging and splitting of trajectories in these methods.

Tracklet Stitching [8] and Linear Programming (LP) based tracking [7] are two other approaches seeking to optimize all trajectories simultaneously over the entire sequence. [8] first generates *tracklets*, which are fragments of tracks formed by conservative grouping of detection responses. The tracklets are then connected by Hungarian partitioning algorithm. This method assumes all tracklets to correspond to true object trajectories and hence is hard to extend to raw detections in each frame where many false alarms are likely to be present. [7] builds a set of subgraphs for every object trajectory with edges between them representing the object interactions. A multi-path search problem on the subgraphs is then solved approximately by linear programming and rounding. It assumes inter-object positions to be relatively stable, and the number of target to be fixed.

3. Our approach

We define data association as a MAP problem. The problem is then mapped into a cost-flow network, and solved with a min-cost flow algorithm. The mapping is based on the observation that there is an analogy between finding non-overlapping object trajectories and finding edge-disjoint paths in a graph; the latter can be solved efficiently by network flow algorithms. We first present the formulation, and then provide the min-cost flow solution.

3.1. MAP under non-overlap constraints

Let $\mathcal{X} = \{\mathbf{x}_i\}$ be a set of object observations, each of which is a detection response, $\mathbf{x}_i = (x_i, s_i, a_i, t_i)$, where x_i is the position, s_i is the scale, a_i is the appearance and t_i is the time step (frame index) of the object. A single trajectory hypothesis is defined as an ordered list of object observations, *i.e.* $T_k = \{\mathbf{x}_{k_1}, \mathbf{x}_{k_2}, \ldots, \mathbf{x}_{k_{l_k}}\}$ where $\mathbf{x}_{k_i} \in \mathcal{X}$. An association hypothesis $\mathcal{T}$ is defined as a set of single trajectory hypotheses, *i.e.* $\mathcal{T} = \{T_k\}$.

The objective of data association is to maximize the posteriori probability of $\mathcal{T}$ given the observation set $\mathcal{X}$:

$$\begin{aligned} \mathcal{T}^* &= \underset{\mathcal{T}}{\mathbf{argmax}}\, P(\mathcal{T}|\mathcal{X}) \\ &= \underset{\mathcal{T}}{\mathbf{argmax}}\, P(\mathcal{X}|\mathcal{T})P(\mathcal{T}) \\ &= \underset{\mathcal{T}}{\mathbf{argmax}} \prod_i P(\mathbf{x}_i|\mathcal{T})P(\mathcal{T}) \end{aligned} \quad (1)$$

assuming that the likelihood probabilities are conditionally independent given the hypothesis $\mathcal{T}$.

It is difficult to optimize Eqn.1 directly, because the space of $\mathcal{T}$ is huge. However, we can reduce the size of the search space by using the observation that one object can only belong to one trajectory. This translates into the constraint that $T_k \in \mathcal{T}$ can not *overlap* with each other, *i.e.*

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

If we further assume that motion of each object is independent, we can decompose Eqn.1 as:

$$\mathcal{T}^* = \underset{\mathcal{T}}{\mathbf{argmax}} \prod_i P(\mathbf{x}_i|\mathcal{T}) \prod_{\mathcal{T}_k \in \mathcal{T}} P(\mathcal{T}_k) \quad (2)$$

$$\text{s.t. } \mathcal{T}_k \cap \mathcal{T}_l = \emptyset, \forall k \neq l \quad (3)$$

The terms in Eqn. (2) are defined as follows:

$$P(\mathbf{x}_i|\mathcal{T}) = \begin{cases} 1-\beta_i & \exists T_k \in \mathcal{T}, \mathbf{x}_i \in \mathcal{T}_k \\ \beta_i & \text{otherwise} \end{cases} \quad (4)$$

$$\begin{aligned} P(\mathcal{T}_k) &= P(\{\mathbf{x}_{k_0}, \mathbf{x}_{k_1}, \ldots, \mathbf{x}_{k_{l_k}}\}) \\ &= P_{entr}(\mathbf{x}_{k_0}) P_{link}(\mathbf{x}_{k_1}|\mathbf{x}_{k_0}) P_{link}(\mathbf{x}_{k_2}|\mathbf{x}_{k_1}) \\ &\quad \ldots P_{link}(\mathbf{x}_{k_{l_k}}|\mathbf{x}_{k_{l_k-1}}) P_{exit}(\mathbf{x}_{k_{l_k}}) \end{aligned} \quad (5)$$

$P(\mathbf{x}_i|\mathcal{T})$ is the likelihood function of observation $\mathbf{x}_i$; a Bernoulli distribution is used to model the cases of an observation being a true detection as well as being a false alarm (β_i is the probability for $\mathbf{x}_i$ being a false alarm). $P(\mathcal{T}_k)$ is modeled as a Markov chain, which includes initialization probability P_{entr}, termination probability P_{exit}, and transition probabilities $P_{link}(\mathbf{x}_{k_{i+1}}|\mathbf{x}_{k_i})$. The precise form of these functions and their estimation from training data are described later in Section 5.

Note that the likelihood function $P(\mathbf{x}_i|\mathcal{T})$ can model not only the observations that are associated in $\mathcal{T}$, *i.e.* true detections, but also those that are not associated, *i.e.* false alarms. This allows the method to select observations, rather than assume all the inputs to be true detections, without additional processing to remove false trajectories after association.

Come back later

3.2. Min-cost flow solution

To couple the non-overlap constraints with the objective function, the following 0-1 indicator variables are defined as

$$f_{en,i} = \begin{cases} 1 & \exists \mathcal{T}_k \in \mathcal{T}, \mathcal{T}_k \text{ starts from } \mathbf{x}_i \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

$$f_{ex,i} = \begin{cases} 1 & \exists \mathcal{T}_k \in \mathcal{T}, \mathcal{T}_k \text{ ends at } \mathbf{x}_i \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

$$f_{i,j} = \begin{cases} 1 & \exists \mathcal{T}_k \in \mathcal{T}, \mathbf{x}_j \text{ is right after } x_i \text{ in } \mathcal{T}_k \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

$$f_i = \begin{cases} 1 & \exists \mathcal{T}_k \in \mathcal{T}, \mathbf{x}_i \in \mathcal{T}_k \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

It's easy to see that these variables are determined for a given association hypothesis $\mathcal{T}$, and vice versa. $\mathcal{T}$ is non-overlap if and only if

$$f_{en,i} + \sum_j f_{j,i} = f_i = f_{ex,i} + \sum_j f_{i,j}, \quad \forall i \quad (10)$$

Next, we incorporate indicators in logarithm of the objective function,

$$\begin{aligned} T &= \operatorname*{argmin}_{\mathcal{T}} \sum_{\mathcal{T}_k \in \mathcal{T}} -\log P(\mathcal{T}_k) + \sum_i -\log P(\mathbf{x}_i|\mathcal{T}) \\ &= \operatorname*{argmin}_{\mathcal{T}} \sum_{\mathcal{T}_k \in \mathcal{T}} (C_{en,k_0} f_{en,k_0} \\ &\quad + \sum_j C_{k_j,k_{j+1}} f_{k_j,k_{j+1}} + C_{ex,k_{l_k}} f_{ex,k_{l_k}}) \\ &\quad + \sum_i (-\log(1-\beta_i) f_i - \log \beta_i (1 - f_i)) \\ &= \operatorname*{argmin}_{\mathcal{T}} \sum_i C_{en,i} f_{en,i} + \sum_{i,j} C_{i,j} f_{i,j} \\ &\quad + \sum_i C_{ex,i} f_{ex,i} + \sum_i C_i f_i \end{aligned} \quad (11)$$

subject to Eqn.10, where

$$C_{en,i} = -\log P_{entr}(\mathbf{x}_i) \qquad C_{ex,i} = -\log P_{exit}(\mathbf{x}_i)$$
$$C_{i,j} = -\log P_{link}(\mathbf{x}_j|\mathbf{x}_i) \qquad C_i = \log \frac{\beta_i}{1-\beta_i}$$

This formulation can be mapped into a cost-flow network $G(\mathcal{X})$ with source s and sink t. Given an observation set

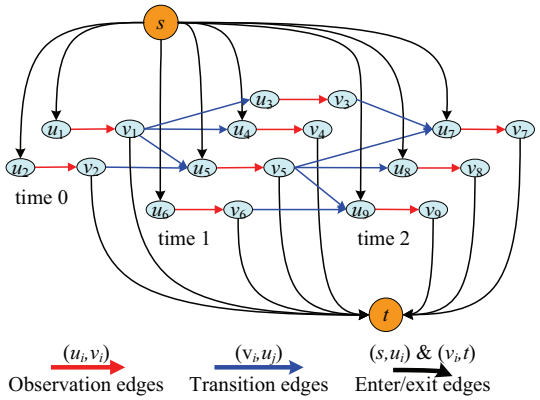


Figure 2. A example of the cost-flow network with 3 timesteps and 9 observations

$\mathcal{X}$: for every observation $\mathbf{x}_i \in \mathcal{X}$, create two nodes u_i, v_i, create an arc (u_i, v_i) with cost $c(u_i, v_i) = C_i$ and flow $f(u_i, v_i) = f_i$, an arc (s, u_i) with cost $c(s, u_i) = C_{en,i}$ and flow $f(s, u_i) = f_{en,i}$, and an arc (v_i, t) with cost $c(v_i, t) = C_{ex,i}$ and flow $f(v_i, t) = f_{ex,i}$. For every transition $P_{link}(\mathbf{x}_j|\mathbf{x}_i) \neq 0$, create an arc (v_i, u_j) with cost $c(v_i, u_j) = C_{i,j}$ and flow $f(v_i, u_j) = f_{i,j}$. An example of such a graph is shown in Figure 2. Eqn.10 is equivalent to the flow conservation constraint and Eqn.11 to the cost of flow in G. Finding optimal association hypothesis $\mathcal{T}^*$ is equivalent to sending the flow from source s to sink t that minimizes the cost.

The cost-flow network formulation is an intuitive representation of multiple object tracking: each flow path can be interpreted as an object trajectory, the amount of the flow sent from s to t is equal to the number of object trajectories, and the total cost of the flow on G corresponds to the log-likelihood of the association hypothesis. The flow conservation constraint guarantees that no flow paths share a common edge, and therefore no trajectories overlap. If all the edge costs in G were positive, the min-cost flow would be the trivial empty zero-cost flow. However, for any observation x_i that is more likely to be a true detection ($\beta_i < 0.5$), the cost C_i of edge (u_i, v_i) is negative; this allows the optimal cost to become below zero by sending flows through these negative-cost edges.

The optimal cost should be calculated over all possible $f(G)$, where $f(G)$ is the amount of flow sent from source to sink. It is known that for a given $f(G)$, the minimal cost can be solved for in polynomial time by a min-cost flow algorithm[11]. The entire optimization process is described as Algorithm 1. It can also be proven that the minimal cost is a convex function w.r.t $f(G)$. Hence the enumeration over all possible $f(G)$ can be replaced by a Fibonacci search, which finds the global minimal cost by at most $O(\log n)$ executions of the min-cost flow algorithm.

Let $n = |\mathcal{X}|$, m be the number of edges in G, which

- Construct the graph $G(V, E, C, f)$ from observation set $\mathcal{X}$
- Start with empty flow
- WHILE ($f(G)$ can be augmented)
  - Augment $f(G)$ by one.
  - Find the min cost flow by the algorithm of [12].
  - IF (current min cost $<$ global optimal cost)
    Store current min-cost assignment as global optimum.
- Return the global optimal flow as the best association hypothesis

Algorithm 1:Find MAP trajectories by min-cost flow.

grows linearly with n. Let K be the number of executions of the min-cost flow algorithm, which is bounded by $\log(n)$. The running time of our method is K times the complexity of the min-cost flow algorithm. One efficient min-cost flow algorithm is the scaling push-relabel method proposed by Goldberg[12] (we use *CS2* implementation from *Andrew Goldberg's Network Optimization Library* at http://www.avglab.com/adrew/soft.html). This algorithm has a worst-case running time of $O(n^2 m \log n)$, but usually takes much less time on real data. We find that the run time grows only linearly with the number of observations as shown in Section 6.4; likely because of the nature of the network where transitions between observations are temporally constrained.

Algorithm 1 provides a general framework for data association. Different from methods which either optimize each trajectory separately or suffer from the combinatorial explosion of the hypotheses space, this method is able to find the global optimum efficiently. Next, we extend our method to track through long-term occlusions.

4. Explicit occlusion model (EOM)

The formulation in Section 3 is capable of tracking short-term missed detections, including those caused by occlusion. However, long-term occlusions, if just treated as missing data, cannot be handled without impairing performance. If we allow association of observations with a large temporal gap between them, the possibility of creating errant association also increases. To effectively track through long-term occlusions, we propose to reason explicitly about which objects may be occluding which others by constructing an Explicit Occlusion Model(EOM). The EOM generates a set of occlusion hypotheses and combines them with the input observations by a set of occlusion constraints. Only occlusions between tracked targets are addressed.

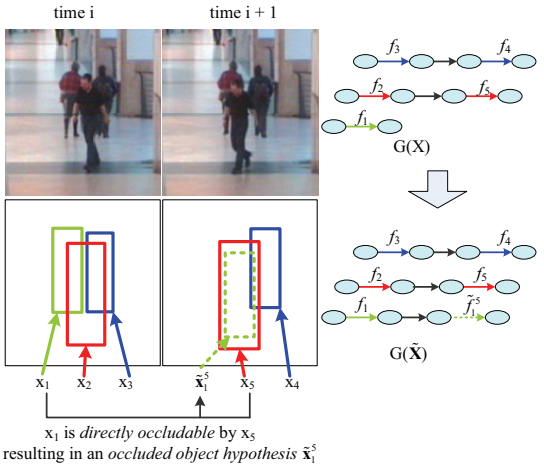


Figure 3. An example of adding *occluded object hypothesis* and corresponding changes in the cost-flow network:the solid rectangles are input observations; the dashed-line rectangle is an *occluded object hypothesis*.

4.1. Occlusion hypotheses and constraints

The first step of EOM is to expand the observation set $\mathcal{X}$ by adding *occluded object hypothesis*.

We say that observation $\mathbf{x}_i$ is *directly occludable* by $\mathbf{x}_j$ if and only if the distance $|x_i - x_j|$ and scale difference s_i/s_j are below certain thresholds, and define the corresponding *occluded object hypothesis* as

$$\tilde{\mathbf{x}}_i^j = (x_j, s_i, a_i, t_j) \tag{12}$$

where x_j and t_j are the position and time step of $\mathbf{x}_j$, and s_i and a_i are the size and appearance of $\mathbf{x}_i$.

For every pair $\{\mathbf{x}_i, \mathbf{x}_j\}$ (or $\{\tilde{\mathbf{x}}_i^k, \mathbf{x}_j\}$) in $\mathcal{X}$ such that $\mathbf{x}_i$ (or $\tilde{\mathbf{x}}_i^k$) is *directly occludable* by $\mathbf{x}_j$, generate a new *occluded object hypothesis* $\tilde{\mathbf{x}}_i^j$ and add it to the observation set $\mathcal{X}$. Repeat this until no new hypotheses can be generated. The procedure of adding hypotheses is illustrated in Figure 3.

Let $\tilde{\mathcal{X}}$ be the observation set obtained by expanding the original $\mathcal{X}$ as above. Note that $|\tilde{\mathcal{X}}| \leq |\mathcal{X}|^2$ after all duplicate hypotheses are removed; the number of observation does not grow exponentially. In practice, hypotheses that are similar in appearance and size are merged by mean-shift clustering to further reduce the size of $\tilde{\mathcal{X}}$.

In the second step of EOM, the proposed MAP formulation(Eqn.11) is applied again to the set $\tilde{\mathcal{X}}$, except for two differences: first, there is no observation likelihood term $P(x|\mathcal{T})$ for any hypothesis $\tilde{\mathbf{x}}$, *i.e.* $P(\tilde{\mathbf{x}}|\mathcal{T}) = 1$; second, a set of occlusion constraints are imposed as

$$\tilde{f}_i^j \leq f_j, \forall\, \tilde{\mathbf{x}}_i^j \in \tilde{\mathcal{X}} \tag{13}$$

where $\tilde{f}_i^j$ is the indicator function for the hypothesis $\tilde{\mathbf{x}}_i^j$. As any f can be only 0 or 1, these constraints guarantee

that an *occluded object hypothesis* can be used ($\tilde{f}_i^j = 1$) in association only if the object that occludes it is also used ($f_j = 1$).

Since the objective function remains unchanged through the extension, we can still optimize Eqn.11, subject to Eqn.10 and the new constraint Eqn.13. We solve the EOM-based data association by using an iterative approach built on Algorithm 1.

4.2. An iterative solution

Algorithm 1 provides an optimal solution when objects are not occluded. To account for occlusions, we take the trajectories found by the Algorithm 1 to be true trajectories and add hypotheses that are occluded by these true trajectories to the network. Algorithm 1 is then applied to the expanded network. This process is repeated to infer occlusions and associations iteratively.

More precisely, first the original MAP formulation with the input observation set $\mathcal{X}$ is solved with the Algorithm 1. Let $\mathcal{T}^*(\mathcal{X}) = \{\mathcal{T}_k^*(\mathcal{X})\}$ be the optimal association hypothesis. For any observation $\mathbf{x}_r \in \cup_k \mathcal{T}_k^*(\mathcal{X})$, the indicator f_r is fixed to be 1. Then *occluded object hypothesis* $\tilde{\mathbf{x}}_i^r$ is generated for any $\mathbf{x}_i$ that is *directly occludable* by $\mathbf{x}_r$. Let the expanded observation set be $\tilde{X} = \mathcal{X} \cup \{\tilde{x}_i^r\}$. Because f_r's are bound to 1, the occlusion constraints for any $\tilde{x}_i^r$ hold automatically. Therefore, instead of having a set of occlusion constraints, we now have

$$f_r \geq 1, \forall \mathbf{x}_r \in \cup_k \mathcal{T}_k^*(\mathcal{X}) \tag{14}$$

Eqn.14 is equivalent to a set of lower bound constraints on the values of $f(u_r, v_r)$ in the cost-flow network. Since the min-cost flow with lower bound constraints can still be solved by the same algorithm [12], Algorithm 1 can be applied again to solve the optimal data association on $\tilde{X}$ with constraints Eqn.14. The procedure of estimating min-cost flow and expanding observation set is repeated until convergence or a preset maximum number of iteration is reached. The approach is described as Algorithm 2 in Table 4.2. In practice, we find that the algorithm usually achieves its optimal performance after two iterations.

Occlusion events can be inferred from the output of Algorithm 2 when an *occluded object hypothesis* is used in the solution, while a missed detection is inferred when there is temporal discontinuity in the association. Based on this, we can infer events of occlusion and missed detection as shown in our results (Figure 1,4,5).

Different from previous works such as [9, 8], which model occlusion through splitting and merging of the trajectories, our method generates occlusion hypotheses explicitly to recover the observations that is missing due to occlusions, and therefore gives a more unified approach because recovered observations are treated in the same way as input observations.

- Let $\mathcal{X}$ be the input observation set
- Let lower bound constraint set $\mathcal{L} = \emptyset$
- DO
  - Solve $\mathcal{T}^*(\mathcal{X})$ using Algorithm 1 with constraint $\mathcal{L}$
  - For each $\mathbf{x}_m \in \mathcal{T}^*(\mathcal{X})$
    add $f_m \geq 1$ to $\mathcal{L}$
    generate $\tilde{\mathbf{x}}_i^m$ for any $\mathbf{x}_i$ *directly occludable* by $\mathbf{x}_m$
  - Let $\mathcal{X} \leftarrow \mathcal{X} \cup \{\tilde{x}_i^m\}$
- WHILE not (converged or max number of iteration reached)
- Return the final optimal flow assignment

Algorithm 2: Find MAP trajectories with occlusion reasoning.

5. Implementation details

In this section, we describe the estimation of the four parameters β_i, P_{entr}, P_{exit} and P_{link} in our framework. As they are directly related to the input observations, they can be estimated from the training data statistically. β_i,P_{entr} and P_{exit} are defined as

$$\beta_i = \text{miss detection rate of the detector} \tag{15}$$

$$P_{entr} = P_{exit} = \frac{\text{number of trajectories}}{\text{number of hypotheses}} \tag{16}$$

As the number of trajectories is data dependent, an EM approach is used to estimate P_{entr} and P_{exit} during the optimization.

The model of P_{link} is defined to employ the information from observations by

$$P_{link}(\mathbf{x}_j|\mathbf{x}_i) = P(s_j|x_i, s_i, \Delta t)P(x_j|x_i, \Delta t) P(a_j|a_i)P(\Delta t) \tag{17}$$

The terms on the right correspond to size, position, appearance and time gap respectively, where conditional independence of the other terms is assumed given time interval Δt, except between scale and position. The position and size terms are assumed to be normal distributions; they are learned from training data.

For the appearance term, two RGB histograms, a_i and a_j are extracted from the detection responses $\mathbf{x}_i$ and $\mathbf{x}_j$ respectively, and $P(a_j|a_i)$ is defined based on the Bhattacharyya distance A_{ij}, *i.e.*

$$P(a_j|a_i) = \frac{N(A_{ij}; A_s, \sigma_s^2)}{N(A_{ij}; A_s, \sigma_s^2) + N(A_{ij}; A_d, \sigma_d^2)} \tag{18}$$

where $N(x; A_s, \sigma_s^2)$ and $N(x; A_d, \sigma_d^2)$ are the normal distributions of A_{ij} between the same object and different objects respectively; they are learned from training data.

The time gap component is defined by an exponential model based on the missing rate α of the detector as

$$P(\Delta t) = \begin{cases} Z_t \alpha^{\Delta t - 1} & 1 \leq \Delta t \leq \xi \\ 0 & \Delta t < 1 \quad \text{or} \quad \Delta t > \xi \end{cases} \tag{19}$$

where ξ is the maximal allowed time gap.

6. Evaluations

In this section, we show results of our method on two datasets: the CAVIAR videos [13] and the ETH Mobile Scene (ETHMS)[14]. Both dataset are very challenging because of the heavy occlusions and poor image contrast from background. Our method is evaluated by its tracking performance, detection performance and speed. Our method is compared with Wu *et al.*'s[2] on CAVIAR, as they have reported the best tracking result on the dataset. On ETHMS it is compared with Ess *et al.*'s detection method[14], as it is the only method evaluated on the dataset. The results show good performance, with fewer false alarms and trajectory fragments than the previous methods. Our method is also more efficient compared to other global methods.

6.1. Experiment settings

The CAVIAR dataset includes 26 video sequences of a walkway in a shopping center taken by a single camera with frame size of 385×288 and frame rate of 25fps. The ETHMS dataset includes 4 video sequences of street scenes taken by a moving camera, with frame size of 640×480 and frame rate of 15fps. Both dataset include many inter-person occlusions in crowded scenes, with poor contrast between objects and background. For CAVIAR, we test on 20 videos (25587 frames total) and use the other 6 videos for training. For ETHMS, we test on sequence #1 (999 frames) and use the other 3 videos for training. The input observation set is from the output files of the human detector by Wu *et al.* [2]. However, we do not make use of the part-based reasoning proposed in [2], but take all the detection responses in the files as our input set. People that are too small in the images(less than 24 pixel in width) or partially out of the scene are not counted in the evaluation.

6.2. Tracking Performance

We evaluate the tracking performance according to the following five metrics proposed in [2]:

- mostly tracked trajectories(MT), the number of trajectories that are successfully tracked for more than 80%;
- mostly lost trajectories(ML), the number of trajectories that are tracked for less than 20%;
- partially tracked trajectories(PT), the number of trajectories that are tracked between 20% and 80%;
- fragmentation(FRMT), the number of times a trajectory is interrupted;
- ID switches(IDS), the number of times two trajectories switch their IDs.

The results on CAVIAR are shown in Table 1, where GT is the number of trajectories in the ground truth.

Method	GT	MT	PT	ML	FRMT	IDS
Wu *et al.* [2]	140	106	25	9	35	17
Algorithm 1	140	104	29	7	58	7
Algorithm 2	140	120	15	5	20	15

Table 1. Comparison of the tracking results on CAVIAR dataset

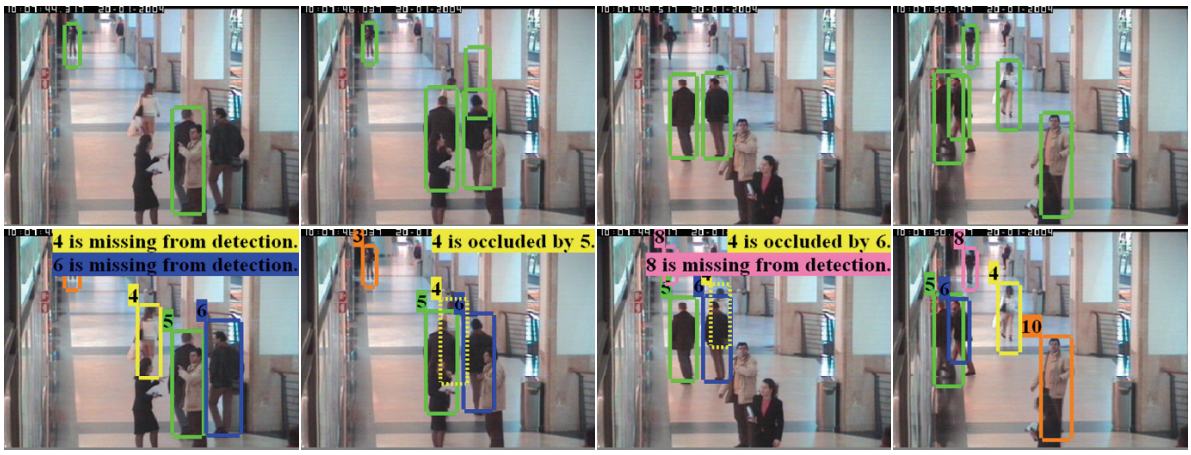
Compared to the result in [2], Algorithm 1 outputs more PT and more FRMT because it has no occlusion model, while EOM-based Algorithm 2 connects a large portion of the trajectory fragments to yield more MT. Some pictorial results are given in Figure 4 to show that the EOM-based Algorithm 2 can recover trajectories from full occlusions. In Figure 4, Object #4 is occluded by object #5 for a long time between frame 301 and 389, but is still tracked by Algorithm 2. Result images also show our method provide reasoning of occlusions and missed detections explicitly. Figure 5 shows that our method can recover trajectories in some complicated cases with many full occlusions.

6.3. Detection Performance

We compare the detection rate and false alarms of our method with the input observation set and previous results [2, 14] in Table 2. It shows that Algorithm 1 increases the detection rate and reduces false alarms significantly compared to the input observation set; Algorithm 2 further improves the detection rate but with a slightly higher false alarm rate. The improvement of Algorithm 2 on detection rate is not significant because the number of fully occluded humans in the entire test set is relatively small. Both Algorithm 1 and 2 give big improvements in the false alarm rates compared to the previous methods, while the detection rates are similar. Because our method does not use additional ways to fill detection gaps such as the mean-shift tracker in [2], it can not recover a trajectory if an object is not detected for many consecutive frames.

6.4. Speed

We have measured execution speed of the method on some CAVIAR videos which typically have several objects to be tracked in each frame, the trajectories are usually long as people walk along the corridor and there are persistent occlusions. The processing time of the object detector is not counted here. Even though the theoretical complexity for the min-cost flow algorithm is polynomial, we find that



Frame 258 Frame 301 Frame 389 Frame 420

Figure 4. Detection inputs and tracking results on CAVIAR[13]: our method can remove false alarms such as in frame 301, recover missed detection such as in frame 258, and track through heavy occlusions such as in frame 301 and 389.

Dataset	Method	Detection rate	False Alarm per frame
CAVIAR	Input observation set	72.8%	0.270
	Wu *et al*. [2]	75.2%	0.281
	Algorithm 1	74.3%	0.081
	Algorithm 2	76.4%	0.105
ETHMS	Input observation set	64.3%	1.54
	Leibe *et al*. [2]	47%	1.5
	Algorithm 1	68.3%	0.85
	Algorithm 2	70.4%	0.97

Table 2. Detection and false Alarm rate on CAVIAR dataset

the complexity grows only linearly with the number of observations in Algorithm 2. In one example, there are 7000 input observations over a sequence of 3500 frames, *i.e*. 140 seconds, Algorithm 1 finds the global optimum in 30 seconds using a 3.7GHz PC. Algorithm 2, applied to the same data, expands the input set with 11000 occluded object hypotheses, and finds a solution in about 2 minutes. Hence our method is real-time and the total time is likely to be dominated by the detection step.

Because of the efficiency of our algorithms, we process every video in CAVIAR dataset globally, without partitioning or using a sliding window as would be necessary for the previous combinatorial algorithms. Sliding window techniques may still be useful if much longer sequences are to be processed or online results are required.

7. Conclusion

We have presented a novel data association framework for multiple object tracking that optimizes the association globally using all the observations from the entire sequence. False alarms, initialization and termination of the trajectory, and inference of occlusions is modeled intrinsically in the method. An optimal solution is provided based on the min-cost network flow algorithms. Though the complexity of the algorithms is polynomial, in practice, we find them to be highly efficient. Experiment results indicate that global data association is helpful, especially for reducing trajectory fragments and improving trajectory consistency, while maintaining efficiency. The framework is general and can be easily adapted to apply to tracking any class of objects for which reasonable detectors are available.

8. Acknowledgement

This research was supported, in part, by the Office of Naval Research under Contract #N00014-06-1-0470. The views expressed here do not necessarily reflect the position or the policy of the United States Government.

References

[1] Z. Khan, T. Balch, and F. Dellaert, "MCMC-based particle filtering for tracking a variable number of interacting targets", *PAMI*, vol. 27, no. 11, 2005. 1, 2

[2] B. Wu and R. Nevatia, "Tracking of multiple, partially occluded humans based on static body part detection", *CVPR*, 2006. 1, 2, 6, 7

[3] D.B. Reid, "An algorithm for tracking multiple targets", *Trans. on Automatic Control*, 1979. 1, 2

[4] Y. Bar-Shalom, T. Fortmann, and M. Scheffe, "Joint probabilistic data association for multiple targets in clutter", in *Information Sciences and Systems*, 1980. 1, 2

[5] J. Berclaz, F. Fleuret, and P. Fua, "Robust people tracking with global trajectory optimization", in *CVPR*, 2006. 1, 2

Figure 5. First and second row are the detection and tracking on ETHMS[14]; fifth to seventh row are tracking on CAVIAR[13].

[6] L. Zhang, B. Wu, and R. Nevatia, "Detection and tracking of multiple humans with extensive pose articulation", in *ICCV*, 2007. 1, 2

[7] H. Jiang, S. Fels, and J. J. Little, "A linear programming approach for multiple object tracking", *CVPR*, 2007. 1, 2

[8] A. G. Amitha Perera, C. Srinivas, A. Hoogs, G. Brooksby, and W. Hu, "Multi-object tracking through simultaneous long occlusions and split-merge conditions", *CVPR*, 2006. 1, 2, 5

[9] Q. Yu, G. Medioni, and I. Cohen, "Multiple target tracking using spatio-temporal markov chain monte carlo data association", *CVPR*, 2007. 1, 2, 5

[10] B. Leibe, K. Schindler, and L. Van Gool, "Coupled detection and trajectory estimation for multi-object tracking", *ICCV*, 2007. 1, 2

[11] R. Ahuja, T. Magnanti, and J.Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, 1993. 3

[12] A. V. Goldberg, "An efficient implementation of a scaling minimum-cost flow algorithm", *J. Algorithms*, 1997. 4, 5

[13] "http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1". 6, 7, 8

[14] A. Ess, B. Leibe, and L. Van Gool, "Depth and appearance for mobile scene analysis", in *ICCV*, 2007. 6, 8