Computer Vision 2
WS 2018/19

Part 11 – Multi-Object Tracking II
04.12.2018

Prof. Dr. Bastian Leibe

RWTH Aachen University, Computer Vision Group
http://www.vision.rwth-aachen.de

Course Outline

- Single-Object Tracking
- Bayesian Filtering
  - Kalman Filters, EKF
  - Particle Filters
- Multi-Object Tracking
  - Introduction
  - MHT, (JPDAF)
  - Network Flow Optimization
- Visual Odometry
- Visual SLAM & 3D Reconstruction
- Deep Learning for Video Analysis

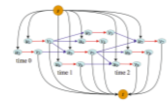
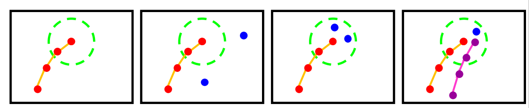


image source: [Zhang, Li, Nevatia, CVPR'08]

Topics of This Lecture

- Recap
  - Track-Splitting Filter
  - MHT
- Data Association as Linear Assignment Problem
  - LAP formulation
  - Greedy algorithm
  - Hungarian algorithm
- Tracking as Network Flow Optimization
  - Min-cost network flow
  - Generalizing to multiple frames
  - Complications
  - Formulation

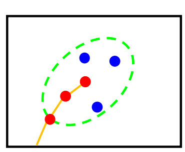
Recap: Motion Correspondence Ambiguities



1. Predictions may not be supported by measurements
   - Have the objects ceased to exist, or are they simply occluded?
2. There may be unexpected measurements
   - Newly visible objects, or just noise?
3. More than one measurement may match a prediction
   - Which measurement is the correct one (what about the others)?
4. A measurement may match to multiple predictions
   - Which object shall the measurement be assigned to?

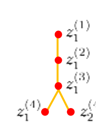
Recap: Mahalanobis Distance

- Gating / Validation volume
  - Our KF state of track $\mathbf{x}_l$ is given by the prediction $\hat{\mathbf{x}}_l^{(k)}$ and covariance $\mathbf{\Sigma}_{p,l}^{(k)}$.
  - We define the innovation that measurement $\mathbf{y}_j$ brings to track $\mathbf{x}_l$ at time k as
    $$\mathbf{v}_{j,l}^{(k)} = (\mathbf{y}_j^{(k)} - \mathbf{x}_{p,l}^{(k)})$$
  - With this, we can write the observation likelihood shortly as
    $$p(\mathbf{y}_j^{(k)} | \mathbf{x}_l^{(k)}) \sim \exp\left\{-\frac{1}{2}\mathbf{v}_{j,l}^{(k)T}\mathbf{\Sigma}_{p,l}^{(k)^{-1}}\mathbf{v}_{j,l}^{(k)}\right\}$$
  - We define the ellipsoidal gating or validation volume as
    $$V^{(k)}(\gamma) = \left\{\mathbf{y} | (\mathbf{y} - \mathbf{x}_{p,l}^{(k)})^T \mathbf{\Sigma}_{p,l}^{(k)^{-1}} (\mathbf{y} - \mathbf{x}_{p,l}^{(k)}) \leq \gamma\right\}$$



Recap: Track-Splitting Filter

- Idea
  - Instead of assigning the measurement that is currently closest, as in the NN algorithm, select the *sequence* of measurements that minimizes the *total* Mahalanobis distance over some interval!
  - Form a track tree for the different association decisions
  - Modified log-likelihood provides the merit of a particular node in the track tree.
  - Cost of calculating this is low, since most terms are needed anyway for the Kalman filter.
- Problem
  - The track tree grows exponentially, may generate a very large number of possible tracks that need to be maintained.

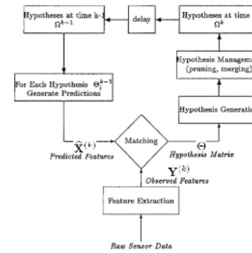


Recap: Pruning Strategies

- In order to keep this feasible, need to apply pruning
  - Deleting unlikely tracks
    - May be accomplished by comparing the modified log-likelihood $\lambda(k)$, which has a χ^2 distribution with kn_z degrees of freedom, with a threshold α (set according to χ^2 distribution tables).
    - Problem for long tracks: modified log-likelihood gets dominated by old terms and responds very slowly to new ones.
    - $\Rightarrow$ Use sliding window or exponential decay term.
  - Merging track nodes
    - If the state estimates of two track nodes are similar, merge them.
    - E.g., if both tracks validate identical subsequent measurements.
  - Only keeping the most likely N tracks
    - Rank tracks based on their modified log-likelihood.

Recap: Multi-Hypothesis Tracking (MHT)

- Ideas
  - Instead of forming a track tree, keep a set of hypotheses that generate child hypotheses based on the associations.
  - Enforce exclusion constraints between tracks and measurements in the assignment.
  - Integrate track generation into the assignment process.
  - After hypothesis generation, merge and prune the current hypothesis set.

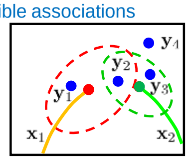


D. Reid, An Algorithm for Tracking Multiple Targets, IEEE Trans. Automatic Control, Vol. 24(6), pp. 843-854, 1979.

Recap: Hypothesis Generation

- Create hypothesis matrix of the feasible associations

$$\Theta = \begin{array}{c} \begin{array}{cccc} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_{fa} & \mathbf{x}_{nt} \end{array} \\ \left[\begin{array}{cccc} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{array}\right] \end{array} \begin{array}{c} \\ \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_3 \\ \mathbf{y}_4 \end{array}$$



- Interpretation
  - Columns represent tracked objects, rows encode measurements
  - A non-zero element at matrix position (i,j) denotes that measurement $\mathbf{y}_i$ is contained in the validation region of track $\mathbf{x}_j$.
  - Extra column $\mathbf{x}_{fa}$ for association as *false alarm*.
  - Extra column $\mathbf{x}_{nt}$ for association as *new track*.
  - Enumerate all *assignments* that are consistent with this matrix.

Recap: Assignments

Z_j	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_{fa}$	$\mathbf{x}_{nt}$
$\mathbf{y}_1$	0	0	1	0
$\mathbf{y}_2$	1	0	0	0
$\mathbf{y}_3$	0	1	0	0
$\mathbf{y}_4$	0	0	0	1

- Impose constraints
  - A measurement can originate from only one object.
  - $\Rightarrow$ Any row has only a single non-zero value.
  - An object can have at most one associated measurement per time step.
  - $\Rightarrow$ Any column has only a single non-zero value, except for $\mathbf{x}_{fa}$, $\mathbf{x}_{nt}$

Recap: Calculating Hypothesis Probabilities

- Probabilistic formulation
  - It is straightforward to enumerate all possible assignments.
  - However, we also need to calculate the probability of each child hypothesis.
  - This is done recursively:

$$\begin{aligned} p(\Omega_j^{(k)}|\mathbf{Y}^{(k)}) &= p(Z_j^{(k)}, \Omega_{p(j)}^{(k-1)}|\mathbf{Y}^{(k)}) \\ &\overset{Bayes}{=} \eta p(\mathbf{Y}^{(k)}|Z_j^{(k)}, \Omega_{p(j)}^{(k-1)}) p(Z_j^{(k)}, \Omega_{p(j)}^{(k-1)}) \\ &= \underset{\text{Normalization factor}}{\eta} \underbrace{p(\mathbf{Y}^{(k)}|Z_j^{(k)}, \Omega_{p(j)}^{(k-1)})}_{\text{Measurement likelihood}} \underbrace{p(Z_j^{(k)}|\Omega_{p(j)}^{(k-1)})}_{\text{Prob. of assignment set}} \underbrace{p(\Omega_{p(j)}^{(k-1)})}_{\text{Prob. of parent}} \end{aligned}$$

Recap: Measurement Likelihood

- Use KF prediction
  - Assume that a measurement $\mathbf{y}_i^{(k)}$ associated to a track $\mathbf{x}_j$ has a Gaussian pdf centered around the measurement prediction $\hat{\mathbf{x}}_j^{(k)}$ with innovation covariance $\hat{\mathbf{\Sigma}}_j^{(k)}$.
  - Further assume that the pdf of a measurement belonging to a new track or false alarm is uniform in the observation volume W (the sensor's field-of-view) with probability W^{-1}.
  - Thus, the measurement likelihood can be expressed as

$$\begin{aligned} p\left(\mathbf{Y}^{(k)}|Z_j^{(k)}, \Omega_{p(j)}^{(k-1)}\right) &= \prod_{i=1}^{M_k} \mathcal{N}\left(\mathbf{y}_i^{(k)}; \hat{\mathbf{x}}_j, \hat{\mathbf{\Sigma}}_j^{(k)}\right)^{\delta_i} W^{-(1-\delta_i)} \\ &= W^{-(N_{fal}+N_{new})} \prod_{i=1}^{M_k} \mathcal{N}\left(\mathbf{y}_i^{(k)}; \hat{\mathbf{x}}_j, \hat{\mathbf{\Sigma}}_j^{(k)}\right)^{\delta_i} \end{aligned}$$

Recap: Probability of an Assignment Set

$$p(Z_j^{(k)}|\Omega_{p(j)}^{(k-1)})$$

- Composed of three terms
  1. Probability of the number of tracks N_{det}, N_{fal}, N_{new}
     - Assumption 1: N_{det} follows a Binomial distribution

$$p(N_{det}|\Omega_{p(j)}^{(k-1)}) = \binom{N}{N_{det}} p_{det}^{N_{det}}(1-p_{det})^{(N-N_{det})}$$

where N is the number of tracks in the parent hypothesis

- Assumption 2: N_{fal} and N_{new} both follow a *Poisson distribution* with expected number of events $\lambda_{fal}W$ and $\lambda_{new}W$

$$p(N_{det}, N_{fal}, N_{new}|\Omega_{p(j)}^{(k-1)}) = \binom{N}{N_{det}} p_{det}^{N_{det}}(1-p_{det})^{(N-N_{det})} \cdot \mu(N_{fal}; \lambda_{fal}W) \cdot \mu(N_{new}; \lambda_{new}W)$$

Recap: Probability of an Assignment Set

2. Probability of a specific assignment of measurements
   - Such that $M_k = N_{det} + N_{fal} + N_{new}$ holds.
   - This is determined as 1 over the number of combinations

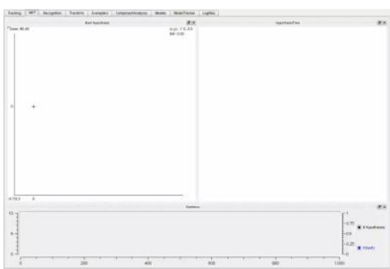
$$\binom{M_k}{N_{det}}\binom{M_k - N_{det}}{N_{fal}}\binom{M_k - N_{det} - N_{fal}}{N_{new}}$$

3. Probability of a specific assignment of tracks
   - Given that a track can be either *detected* or not *detected*.
   - This is determined as 1 over the number of assignments

$$\frac{N!}{(N-N_{det})!}\binom{N-N_{det}}{N_{det}}$$

$\Rightarrow$ When combining the different parts, many terms cancel out!

Laser-based Leg Tracking using Hypothesis Tree MHT



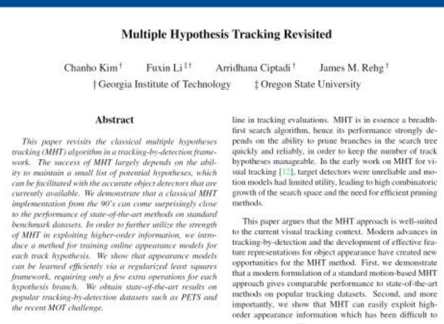
K. Arras, S. Grzonka, M. Luber, W. Burgard, Efficient People Tracking in Laser Range Data using a Multi-Hypothesis Leg-Tracker with Adaptive Occlusion Probabilities, ICRA'08.

video source: Social Robotics Lab, Univ. Freiburg

Laser-based People Tracking using MHT

video source: Social Robotics Lab, Univ. Freiburg

Recent Successes [Kim CVPR'15]



Multiple Hypothesis Tracking Revisited

Chanho Kim † Fuxin Li ‡† Arridhana Ciptadi † James M. Rehg †
† Georgia Institute of Technology ‡ Oregon State University

Abstract

This paper revisits the classical multiple hypotheses tracking (MHT) algorithm in a tracking-by-detection framework. The success of MHT largely depends on the ability to maintain a small list of potential hypotheses, which can be facilitated with the accurate object detectors that are currently available. We demonstrate that a classical MHT implementation from the 90's can come surprisingly close to the performance of state-of-the-art methods on standard benchmark datasets. In order to further utilize the strength of MHT in exploiting higher-order information, we introduce a method for training online appearance models for each track hypothesis. We show that appearance models can be learned efficiently via a regularized least squares framework, requiring only a few extra operations for each hypothesis branch. We obtain state-of-the-art results on popular tracking-by-detection datasets such as PETS and the recent MOT challenge.

line in tracking evaluations. MHT is in essence a breadth-first search algorithm, hence its performance strongly depends on the ability to prune branches in the search tree quickly and reliably, in order to keep the number of track hypotheses manageable. In the early work on MHT for visual tracking [12], target detectors were unreliable and motion models had limited utility, leading to high combinatoric growth of the search space and the need for efficient pruning methods.

This paper argues that the MHT approach is well-suited to the current visual tracking context. Modern advances in tracking-by-detection and the development of effective feature representations for object appearance have created new opportunities for the MHT method. First, we demonstrate that a modern formulation of a standard motion-based MHT approach gives comparable performance to state-of-the-art methods on popular tracking datasets. Second, and more importantly, we show that MHT can easily exploit high-order appearance information which has been difficult to

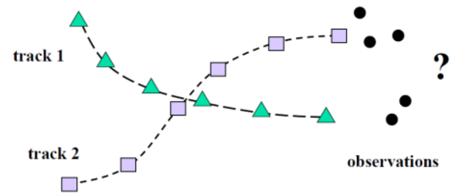
Topics of This Lecture

- Recap
  - Track-Splitting Filter
  - MHT
- Data Association as Linear Assignment Problem
  - LAP formulation
  - Greedy algorithm
  - Hungarian algorithm
- Tracking as Network Flow Optimization
  - Min-cost network flow
  - Generalizing to multiple frames
  - Complications
  - Formulation

Back to Data Association...

- Goal: Match detections across frames

Data Association



- Main question here
  - How to determine which measurements to add to which track?
  - Today: consider this as a matching problem

Linear Assignment Formulation

- Form a matrix of pairwise similarity scores
- Similarity could be
  - based on motion prediction
  - based on appearance
  - based on both

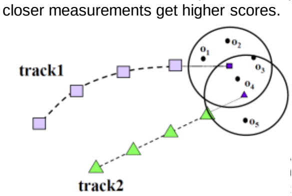
Frame $t+1$ (columns), Frame t (rows)

			
	0.11	0.95	0.23
	0.85	0.25	0.89
	0.90	0.12	0.81

- Goal
  - Choose one match from each row and column to maximize the sum of scores

Linear Assignment Formulation

- Example: Similarity based on motion prediction
  - Predict motion for each trajectory and assign scores for each measurement based on inverse (Mahalanobis) distance, such that closer measurements get higher scores.



	ai1	ai2
1	3.0	
2	5.0	
3	6.0	1.0
4	9.0	8.0
5		3.0

  - Choose at most one match in each row and column to maximize sum of scores

Linear Assignment Problem

- Formal definition
  - Maximize $\sum_{i=1}^{N}\sum_{j=1}^{M} w_{ij} z_{ij}$

    subject to
    $$\left.\begin{array}{l} \sum_{j=1} z_{ij} = 1;\ i = 1, 2, \ldots, N \\ \sum_{i=1} z_{ij} = 1;\ j = 1, 2, \ldots, M \\ z_{ij} \subset \{0, 1\} \end{array}\right\}$$
    Those constraints ensure that Z is a permutation matrix
  - The permutation matrix constraint ensures that we can only match up one object from each row and column.
  - Note: Alternatively, we can minimize cost rather than maximizing weights. $\arg\min_{z_{ij}} \sum_{i=1}^{N}\sum_{j=1}^{M} c_{ij} z_{ij}$

Greedy Solution to LAP

	1	2	3	4	5
1	0.95	0.76	0.62	0.41	0.06
2	0.23	0.46	0.79	0.94	0.35
3	0.61	0.02	0.92	0.92	0.81
4	0.49	0.82	0.74	0.41	0.01
5	0.89	0.44	0.18	0.89	0.14

- Greedy algorithm
  - Find the largest score
  - Remove scores in same row and column from consideration
  - Repeat
- Result: score =

Greedy Solution to LAP

	1	2	3	4	5
1	$\boxed{0.95}$	~~0.76~~	~~0.62~~	~~0.41~~	~~0.06~~
2	~~0.23~~	~~0.46~~	~~0.79~~	$\boxed{0.94}$	~~0.35~~
3	~~0.61~~	~~0.02~~	$\boxed{0.92}$	~~0.92~~	~~0.81~~
4	~~0.49~~	$\boxed{0.82}$	~~0.74~~	~~0.41~~	~~0.01~~
5	~~0.89~~	~~0.44~~	~~0.18~~	~~0.89~~	$\boxed{0.14}$

- Greedy algorithm
  - Find the largest score
  - Remove scores in same row and column from consideration
  - Repeat
- Result: score $=$ 0.95 + 0.94 + 0.92 + 0.82 + 0.14 $=$ 3.77

Is this the best we can do?

Greedy Solution to LAP

	1	2	3	4	5
1	$\boxed{0.95}$	0.76	0.62	0.41	0.06
2	0.23	0.46	0.79	$\boxed{0.94}$	0.35
3	0.61	0.02	$\boxed{0.92}$	0.92	0.81
4	0.49	$\boxed{0.82}$	0.74	0.41	0.01
5	0.89	0.44	0.18	0.89	$\boxed{0.14}$

Greedy solution
score $= 3.77$

	1	2	3	4	5
1	$\boxed{0.95}$	0.76	0.62	0.41	0.06
2	0.23	0.46	$\boxed{0.79}$	0.94	0.35
3	0.61	0.02	0.92	0.92	$\boxed{0.81}$
4	0.49	$\boxed{0.82}$	0.74	0.41	0.01
5	0.89	0.44	0.18	$\boxed{0.89}$	0.14

Optimal solution
score $= 4.26$

- Discussion
  - Greedy method is easy to program, quick to run, and yields “pretty good” solutions in practice.
  - But it often does not yield the optimal solution.

Optimal Solution

- Hungarian Algorithm
  - There is an algorithm called Kuhn-Munkres or “Hungarian” algorithm specifically developed to efficiently solve the linear assignment problem.
  - Reduces assignment problem to bipartite graph matching.
  - When starting from an $N \times N$ matrix, it runs in $\mathcal{O}(N^3)$.
  - $\Rightarrow$ If you need LAP, you should use this algorithm.
- In the following
  - Look at other algorithms that generalize to multi-frame (>2 frames) problems.
  - $\Rightarrow$ Min-Cost Network Flow

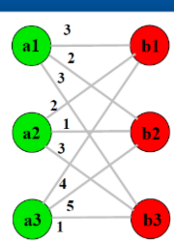
Topics of This Lecture

- Recap
  - Track-Splitting Filter
  - MHT
- Data Association as Linear Assignment Problem
  - LAP formulation
  - Greedy algorithm
  - Hungarian algorithm
- Tracking as Network Flow Optimization
  - Min-cost network flow
  - Generalizing to multiple frames
  - Complications
  - Formulation

Min-Cost Flow

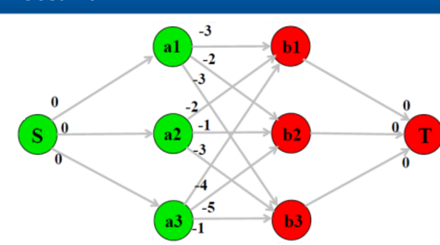
- Small example

	1	2	3
1	3	2	3
2	2	1	3
3	4	5	1



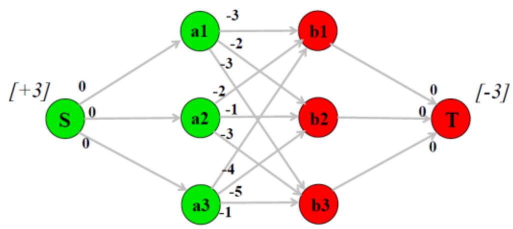
- Network Flow formulation
  - Reformulate Linear Cost Assignment into a min-cost flow problem

Min-Cost Flow



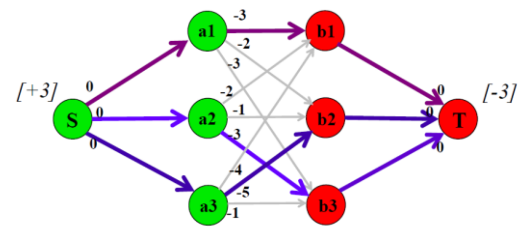
- Conversion into flow graph
  - Transform weights into costs $c_{ij} = \alpha - w_{ij}$
  - Add source/sink nodes with 0 cost.
  - Directed edges with a capacity of 1.

Min-Cost Flow



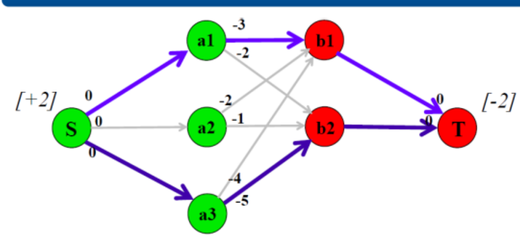
- Conversion into flow graph
  - Pump N units of flow from source to sink.
  - Internal nodes pass on flow ($\sum$ flow in $=$ $\sum$ flow out).
  - $\Rightarrow$ Find the optimal paths along which to ship the flow.

Min-Cost Flow



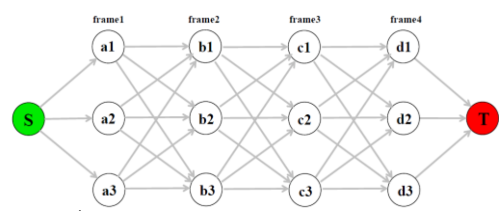
- Conversion into flow graph
  - Pump N units of flow from source to sink.
  - Internal nodes pass on flow ($\sum$ flow in $=$ $\sum$ flow out).
  - $\Rightarrow$ Find the optimal paths along which to ship the flow.

Min-Cost Flow



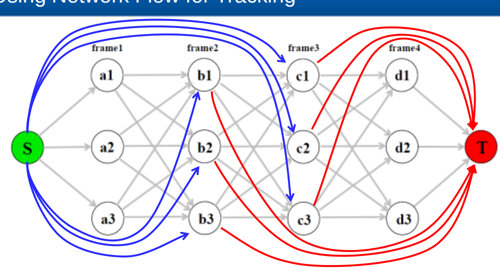
- Nice property
  - Min-cost formalism readily generalizes to matching sets with unequal sizes.

Using Network Flow for Tracking



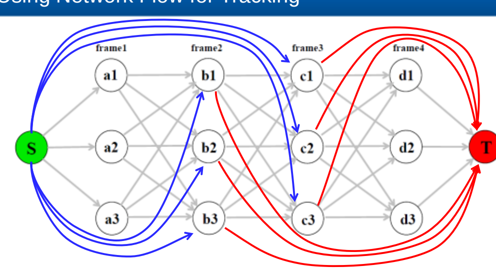
- Approach
  - Seek a globally optimal solution by considering observations over all frames in "batch mode".
    - $\Rightarrow$ Extend two-frame min-cost formulation by adding observations from all frames into the network.

Using Network Flow for Tracking



- Complication 1
  - Tracks can start later than frame1 (and end earlier than frame4)
  - $\Rightarrow$ Connect the source and sink nodes to all intermediate nodes.

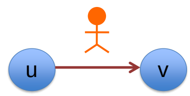
Using Network Flow for Tracking



- Complication 2
  - Trivial solution: zero cost flow!

Using Network Flow for Tracking

- Solution
  - Divide each detection into 2 nodes

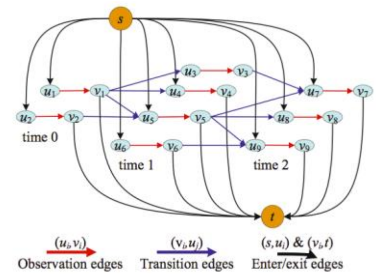


Detection edge

$$C_i = \log \frac{\beta_i}{1-\beta_i} \longleftarrow$$ Probability that detection i is a false alarm

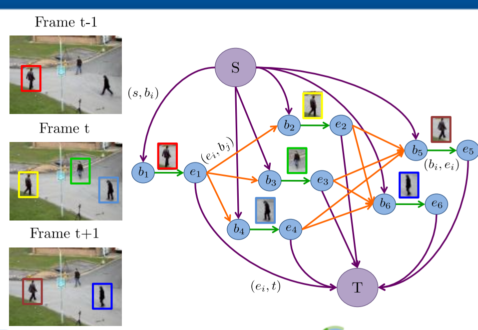
Zhang, Li, Nevatia, Global Data Association for Multi-Object Tracking using Network Flows, CVPR'08.

Network Flow Approach



Zhang, Li, Nevatia, Global Data Association for Multi-Object Tracking using Network Flows, CVPR'08.

Network Flow Approach: Illustration



Min-Cost Formulation

- Objective Function

$$\mathcal{T}* = \underset{\mathcal{T}}{\textbf{argmin}} \sum_i C_{in,i} f_{in,i} + \sum_i C_{i,out} f_{i,out}$$
$$+ \sum_{i,j} C_{i,j} f_{i,j} + \sum_i C_i f_i$$

- subject to
  - Flow conservation at all nodes

$$f_{in,i} + \sum_j f_{j,i} = f_i = f_{out,i} + \sum_j f_{i,j} \; \forall i$$

  - Edge capacities

$$f_i \leq 1$$

Min-Cost Formulation

- Objective Function

$$\mathcal{T}* = \underset{\mathcal{T}}{\textbf{argmin}} \sum_i C_{in,i} f_{in,i} + \sum_i C_{i,out} f_{i,out}$$
$$+ \sum_{i,j} C_{i,j} f_{i,j} + \sum_i C_i f_i$$

$$\downarrow \quad C_i = -log(P_i)$$

- Equivalent to Maximum A-Posteriori formulation

$$\mathcal{T}* = \underset{\mathcal{T}}{\textbf{argmax}} \prod_i P(\mathbf{o}_i|\mathcal{T}) P(\mathcal{T})$$

Min-Cost Formulation

- Objective Function

IN OUT

$$\mathcal{T}* = \underset{\mathcal{T}}{\textbf{argmin}} \boxed{\sum_i C_{in,i} f_{in,i} + \sum_i C_{i,out} f_{i,out}}$$
$$+ \boxed{\sum_{i,j} C_{i,j} f_{i,j}} + \boxed{\sum_i C_i f_i}$$

TRANSITION

Likelihood of the detection

- Equivalent to Maximum A-Posteriori formulation

$$\mathcal{T}* = \underset{\mathcal{T}}{\textbf{argmax}} \boxed{\prod_i P(\mathbf{o}_i|\mathcal{T})} P(\mathcal{T})$$
$$P(\mathcal{T}) = \prod_{T_k \in \mathcal{T}} P(T_k)$$

Independence assumption + Markov

Network Flow Solutions

- Push-relabel method
  - Zhang, Li, Nevatia, Global Data Association for Multi-Object Tracking using Network Flows, CVPR'08.
- Successive shortest path algorithm
  - Berclaz, Fleuret, Turetken, Fua, Multiple Object Tracking using K-shortest Paths Optimization, IEEE PAMI, Sep 2011. (code)
  - Pirsiavash, Ramanan, Fowlkes, Globally Optimal Greedy Algorithms for Tracking a Variable Number of Objects, CVPR'11.
  - These both include approximate dynamic programming solutions

Summary

- Tracking as network flow optimization
- Pros
  - Clear algorithmic framework, equivalence to probabilistic formulation
  - Well-understood LP optimization problem, efficient algorithms available
  - Globally optimal solution
- Cons / Limitations
  - Only applicable to restricted problem setting due to LP formulation
    - Not possible to encode exclusion constraints between detections (e.g., to penalize physical overlap)
    - Motion model can only draw upon information from pairs of detections (i.e., only zero-velocity model possible, no constant velocity models)
  - C_{in} and C_{out} cost terms are quite fiddly to set in practice
    - Too low $\Rightarrow$ fragmentations, too high $\Rightarrow$ ID switches

References and Further Reading

- The original network flow tracking paper
  - Zhang, Li, Nevatia, Global Data Association for Multi-Object Tracking using Network Flows, CVPR'08.
- Extensions and improvements
  - Berclaz, Fleuret, Turetken, Fua, Multiple Object Tracking using K-shortest Paths Optimization, IEEE PAMI, Sep 2011. (code)
  - Pirsiavash, Ramanan, Fowlkes, Globally Optimal Greedy Algorithms for Tracking a Variable Number of Objects, CVPR'11.
- A recent extension to incorporate social walking models
  - L. Leal-Taixe, G. Pons-Moll, B. Rosenhahn, Everybody Needs Somebody: Modeling Social and Grouping Behavior on a Linear Programming Multiple People Tracker, ICCV Workshops 2011.