Computer Vision II - Lecture 12
Multi-Object Tracking II
01.07.2014

Bastian Leibe
RWTH Aachen
http://www.vision.rwth-aachen.de
leibe@vision.rwth-aachen.de

Course Outline
• Single-Object Tracking
• Bayesian Filtering
  - Kalman filters
  - Particle filters
  - Case studies
• Multi-Object Tracking
  - Introduction
  - MHT, JPDAF
  - Network Flow Optimization
• Articulated Tracking

Topics of This Lecture
• Recap: Track-Splitting Filter
  - Motivation
  - Ambiguities
• Multi-Hypothesis Tracking (MHT)
  - Basic idea
  - Hypothesis Generation
  - Assignment
  - Measurement Likelihood
  - Practical considerations

Let’s Formalize This
• Multi-Object Tracking problem
  - We represent a track by a state vector $\mathbf{x}$, e.g.,
    $\mathbf{x} = [x, y, v_x, v_y]^T$
  - As the track evolves, we denote its state by the time index $k$:
    $\mathbf{x}^{(k)} = [x^{(k)}, y^{(k)}, v_x^{(k)}, v_y^{(k)}]^T$
  - At each time step, we get a set of observations (measurements)
    $\mathbf{y}^{(k)} = \{y_1^{(k)}, \ldots, y_{\text{num}}^{(k)}\}$
  - We now need to make the data association between tracks
    $\{x_1^{(k)}, \ldots, x_{\text{num}}^{(k)}\}$ and observations
    $\{y_1^{(k)}, \ldots, y_{\text{num}}^{(k)}\}$:
    $z_j^{(k)} = \text{index of } y_j^{(k)}$ associated with $x_j^{(k)}$

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: Reducing Ambiguities
• Gating
  - Only consider measurements within a certain area around the predicted location.
  ⇒ Large gain in efficiency, since only a small region needs to be searched
• Nearest-Neighbor Filter
  - Among the candidates in the gating region, only take the one closest to the prediction $x_j^{(k)}$
    $z_j^{(k)} = \arg \min_{y_j^{(k)}} (x_j^{(k)} - y_j^{(k)})^T \Sigma_{x_j^{(k)}}^{-1} (x_j^{(k)} - y_j^{(k)})$
  - Better: the one most likely under a Gaussian prediction model
    $z_j^{(k)} = \arg \max_y N(y_j^{(k)} | x_j^{(k)}, \Sigma_{x_j^{(k)}})$
    which is equivalent to taking the Mahalanobis distance
    $z_j = \arg \min_{y_j^{(k)}} (x_j^{(k)} - y_j)^T \Sigma_{x_j^{(k)}}^{-1} (x_j^{(k)} - y_j)$
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 $\lambda(A)$, which has a $\chi^2$ distribution with $kn$ degrees of freedom, with a threshold $\alpha$ (set according to $\chi^2$ distribution tables).
    - Problem for long tracks: modified log-likelihood gets dominated by old terms and responds very slowly to new ones.
  - 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 $\frac{1}{4}$ tracks
    - Rank tracks based on their modified log-likelihood.

Summary: Track-Splitting Filter

- **Properties**
  - Very old algorithm
  - Improvement over NN assignment.
  - Assignment decisions are delayed until more information is available.

- **Many problems remain**
  - Exponential complexity, heuristic pruning needed.
  - Merging of track nodes is necessary, because tracks may share measurements, which is physically unrealistic.
  - Would need to add exclusion constraints such that each measurement may only belong to a single track.
  - Impossible in this framework...

Multi-Hypothesis Tracking (MHT)

- **Ideas**
  - Again associate sequences of measurements.
  - Evaluate the probabilities of all association hypotheses.
  - For each sequence of measurements (a hypothesized track), a standard KF yields the state estimate and covariance

- **Differences to Track-Splitting Filter**
  - Instead of forming a track tree, keep a set of hypotheses that generate child hypotheses based on the associations.
  - After each hypothesis generation step, merge and prune the current hypothesis set to keep the approach feasible.
  - Integrate track generation into the assignment process.


Topics of This Lecture

- Recap: Track-Splitting Filter
  - Motivation
  - Ambiguities
- Multi-Hypothesis Tracking (MHT)
  - Basic idea
  - Hypothesis Generation
  - Assignment
  - Measurement Likelihood
  - Practical considerations

Target vs. Measurement Orientation

- **Target-oriented approaches**
  - Evaluate the probability that a measurement belongs to an established target.

- **Measurement-oriented approaches**
  - Evaluate the probability that an established target or a new target gave rise to a certain measurement sequence.
  - This makes it possible to include track initiation of new targets within the algorithmic framework.

- **MHT**
  - Measurement-oriented
  - Handles track initialization and termination
Challenge: Exponential Complexity

- **Strategy**
  - Generate all possible hypotheses and then depend on pruning these hypotheses to avoid the combinatorial explosion.
  - ⇒ Exhaustive search
    - Tree data structures are used to keep this search efficient

- **Commonly used pruning techniques**
  - Clustering to reduce the combinatorial complexity
  - Pruning of low-probability hypotheses
  - N-scan pruning
  - Merging of similar hypotheses

Hypothesis Generation

- **Formalization**
  - Set of hypotheses at time $k$: $\Omega^{(k)} = \{\omega^{(k)}_j\}$
  - This set is obtained from $\Omega^{(k-1)}$ and the latest set of measurements $Y^{(k)} = \{y^{(k)}_1, \ldots, y^{(k)}_m\}$
  - The set $\Omega^{(k)}$ is generated from $\Omega^{(k-1)}$ by performing all feasible associations between the old hypotheses and the new measurements $Y^{(k)}$.

- **Feasible associations** can be
  - A continuation of a previous track
  - A false alarm
  - A new target

Assignments

- **Turning feasible associations into assignments**
  - For each feasible association, we generate a new hypothesis.
  - Let $f_j^{(k)}$ be the $j$-th hypothesis at time $k$ and $f^{(k-1)}_{p(j)}$ be the parent hypothesis from which $f_j^{(k)}$ was derived.
  - Let $\mathcal{Z}_{j}^{(k)}$ denote the set of assignments that gives rise to $f_j^{(k)}$.
  - Assignments are again best visualized in matrix form

<table>
<thead>
<tr>
<th>$Z_j$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_{fa}$</th>
<th>$x_{nt}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_1$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$y_2$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$y_3$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$y_4$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Assignments

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

\[
p(\mathbf{Y}^{(k)}, \mathbf{Z}_{j}^{(k)}) = n_{\text{Edges}}(\mathbf{Y}^{(k)}, \mathbf{Z}_{j}^{(k)}) p(\mathbf{Y}^{(k)}, \mathbf{Z}_{j}^{(k)})
\]

- Measurement likelihood

\[
p(\mathbf{Y}^{(k)} | \mathbf{Z}_{j}^{(k)}) = 1/(N! \cdot \lambda_{\text{det}}^{N_{\text{det}}} \lambda_{\text{fal}}^{N_{\text{fal}}} \lambda_{\text{new}}^{N_{\text{new}}})
\]

- Prob. of parent

\[
p(\mathbf{Y}^{(k)}) = 1/(W^{N_{\text{det}} + N_{\text{fal}} + N_{\text{new}}})
\]

- Prob. of assignment set

\[
p(\mathbf{Z}_{j}^{(k)}) = \frac{M_{j}}{W^{N_{\text{det}} + N_{\text{fal}} + N_{\text{new}}}}
\]

- Normalization factor

\[
p(\mathbf{Y}^{(k)}) = \prod_{i=1}^{N} N_{\text{det}}(\mathbf{y}_{i}^{(k)}; \mathbf{x}_{j}, \Sigma_{j}^{(k)})^{N_{\text{det}}} W^{-1 - N_{\text{det}}}
\]

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 \( \mathbf{z}_{j}^{(k)} \) with innovation covariance \( \Sigma_{j}^{(k)} \).
  - 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

\[
p(\mathbf{Y}^{(k)} | \mathbf{Z}_{j}^{(k)}) = \prod_{i=1}^{N} N_{\text{det}}(\mathbf{y}_{i}^{(k)}; \mathbf{x}_{j}, \Sigma_{j}^{(k)})^{N_{\text{det}}} W^{-1 - N_{\text{det}}}
\]

- Probability of an Assignment Set

\[
p(\mathbf{Z}_{j}^{(k)}) = \frac{M_{j}}{W^{N_{\text{det}} + N_{\text{fal}} + N_{\text{new}}}}
\]

References and Further Reading

- A good tutorial on Data Association

- Reid’s original MHT paper