Models and Filters for camera-based Multi-target Tracking

Dr.-Ing. Mirko Meuter
interact!Ve Summer School
4-6 July, 2012
Outline: Contents of the Presentation

- From detection to tracking
- Overview over camera based multi-target tracking systems
- Association of measurements to tracks
- Filters for different tracking systems
  - Kalman Filter
  - EKF, UKF
  - Particle Filter
  - IMM Filter
- Models for camera-based tracking
  - 3D backprojection measurement model
  - Lane tracking models
  - Traffic sign tracking model
  - Pedestrian tracking model
  - Vehicle tracking models
From Detection to Tracking

Objective
• Follow objects over time
• Filter object trajectories
• Reduce failures and noise
Structure of a camera-based Tracking System

- **Input**
  - Single frame recognition
  - Noisy
  - No temporal connection

- **Output**
  - Object lists ("Track Lists")
  - Estimated object states
Structure of a camera-based Tracking System

- System
  - Measurement association
  - Creation and deletion of tracks
  - Filter interaction

- Filtering
  - Requires modeling of the object motion and the measurement process
  - Motion extraction
  - Noise reduction
Association of Tracks to Measurements

• Input for classic tracking filters is a temporal list of measurement readings
• The measurements must be assigned to each filter / filter element
Association of Tracks to Measurements

- Gating can be used to reduce the number of feasible association sets

- The expected / predicted measurement innovation covariance can be used to determine the association area (KF)
Association of Tracks to Measurements

- Assign measurements to tracks based on the position / velocity / size
  - Evaluate least overall distance / highest probability of the association set

- Hard associations
  - Assign one measurement to one track
  - E.g. Global nearest neighbour / HM

- Soft associations
  - Assign all measurements in the environment with a certain probability
  - E.g. PDA, JPDA and extensions

- Association based on a delayed decision
  - Maintain a set of association hypothesis
  - E.g. MHT and extensions

- Association using image cues
  - Association based on feature similarity
  - E.g. comparison of descriptor vectors used for interest point tracking
Filter Model

Objective
• Filter a set of temporal measurement readings \((z)\)
• Extract object state information \((x)\)

Model
• Object evolves according to a stochastic markov process
  \[ p(x_k|x_{k-1}, \ldots, x_1) = p(x_k|x_{k-1}) \]
• Stochastic measurement process

Solution
• Use a recursive Bayesian filter to estimate the probability density of \(x\) conditioned on all measurements
• Extract \(x\)

\[ p(x_k|Z_k) \]

\(Z_{k-1}\)
\(Z_k\)

\(X_{k-1}\)
\(X_k\)
Tracking Filters for Linear Systems

- **Kalman Filter**
  - Linear system and measurement model
  - Normally distributed system and measurement noise
  - Normally distributed state pdf
  - Estimates mean and covariance
Tracking Filters for Linear Systems

• Fixed Gain Kalman Filter
  • Simplification of the Kalman filter for constant noise and system matrices
  • Steady state gain is used to map measurement deviations to the state space
  • No propagation of the covariance matrix necessary
• Alpha-beta Filter
  • Simplified KF with constant velocity model
  • Simplified parameterization
  • No covariance propagation
• Alpha-beta-gamma Filter
  • Simplified KF with constant acceleration model
  • Simplified parameterization
  • No covariance propagation
Tracking Filters for Linear Systems

• Alpha-beta filter

\[ p_t = p_{t-1} + v \, dt \]

\[
\begin{bmatrix}
\hat{p}_{k|k-1} \\
\hat{v}_{k|k-1}
\end{bmatrix}
= \begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix}
\begin{bmatrix}
\hat{p}_{k-1|k-1} \\
\hat{v}_{k-1|k-1}
\end{bmatrix}
+ \begin{bmatrix}
\alpha(p - \hat{p}_{k|k-1}) \\
\beta \frac{dt}{dt} (p - \hat{p}_{k|k-1})
\end{bmatrix}
\]

• Alpha-beta-gamma filter

\[ p_t = p_{t-1} + v \, dt + \frac{1}{2} a dt^2 \]

\[
\begin{bmatrix}
\hat{p}_{k|k-1} \\
\hat{v}_{k|k-1} \\
\hat{a}_{k|k-1}
\end{bmatrix}
= \begin{bmatrix} 1 & dt & \frac{1}{2} dt^2 \\ 0 & 1 & dt \\ 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix}
\hat{p}_{k-1|k-1} \\
\hat{v}_{k-1|k-1} \\
\hat{a}_{k-1|k-1}
\end{bmatrix}
+ \begin{bmatrix}
\alpha(p - \hat{p}_{k|k-1}) \\
\beta \frac{dt}{dt} (p - \hat{p}_{k|k-1}) \\
\gamma \frac{2 dt^2}{2 dt^2} (p - \hat{p}_{k|k-1})
\end{bmatrix}
\]

4 to 6 lines of code and the filter is ready
Tracking Filters for Nonlinear Systems

Extended Kalman filter

- Normally distributed state and noise pdfs
- Linearizes system and measurement equations around the mean estimate
- A relinearization around the estimate can be used to improve the results
Tracking Filters for Nonlinear Systems

- Unscented Kalman filter
  - Normally distributed state and noise pdfs
  - Propagates a small set of sigma points through the nonlinear function
  - Better Covariance estimates
  - Higher calculation effort (except for special forms)
Tracking Filters for Nonlinear Systems

• Particle filter
  • General Bayesian filter
  • Approximates the probability densities using a number of weighted points in the state space
  • Can be used for all kinds of system and noise models
  • Can be computationally expensive

• Further Alternatives
  • GMM filters, Rao blackwellized Kalman filters, …)
Coupling of Several Hypotheses

The Interacting Multiple Model Filter

- Interacting Multiple Model Filter (IMM-Filter)
  - Filter bank of filters with different properties
  - Automatic probabilistic mode change

- Can be used for:
  - Dynamic system noise adaption for maneuver situations
  - Model switching to switch to less complex models
    - E.g. switching from a constant acceleration to a constant velocity model
  - Model switching to avoid observability problems
  - Data fusion to fuse sensors with a small probability for false readings
  - High filter performance and robustness
Coupling of Several Hypotheses
The Interacting Multiple Model Filter

IMM Filter Mixing

Filter 1

Filter 2

Filter N

Calculate Probabilities

Output Mixing

$\mu_{01}$

$\mu_{0n}$

$\hat{x}_{01}$

$\hat{P}_{01}$

$\hat{y}_{01}$

$\hat{y}_{02}$

$\hat{y}_{0n}$

$\hat{x}_{02}$

$\hat{P}_{02}$

$\hat{x}_{0n}$

$\hat{P}_{0n}$

$\mu_{11}$

$\mu_{12}$

$\mu_{1n}$

$\hat{x}_{11}$

$\hat{P}_{11}$

$\hat{x}_{12}$

$\hat{P}_{12}$

$\hat{x}_{1n}$

$\hat{P}_{1n}$

$\hat{x}_{1}$

$\hat{P}_{1}$

$\hat{y}_{1}$
Models for Camera Applications

• 2D Models
  • Image plane motion
  • Often constant velocity or constant acceleration model
  • No direct relation to real world motion
  • Simple and fast

• 3D models
  • World motion model
    Object specific
    • Object type is known
  • Ego motion compensation
    • velocity and yaw rate data
  • Projective measurement model
    • Requires camera calibration data
3D Measurement Model
Relation between 3D World and 2D Image Coordinates

• In a monocular camera, the distance information is lost
• For distance reconstruction, some constraints are needed
  • Observed movement of stationary(!) points and ego motion estimation (Slam) or use of sensor ego motion information
  • Size observations and size constraints of objects
  • Ground plane assumption
• Otherwise, the distance is unobservable!

• For vehicles, camera pitch should be taken into account in the model or compensated
The coefficients of the camera calibration matrix completely capture the intrinsic and extrinsic parameters of the camera.

Monocular Model-Based 3D Tracking of Rigid Objects: A Survey
Vincent Lepetit and Pascal Fua
Ego Motion Compensation

- Simple ego motion model
  - Uses velocity and yaw rate / steering angle
  - Does not include the sideslip angle
  - No consideration of dynamic effects (e.g. understeering or oversteering)

\[
x_e(t) = \frac{v_e}{\varphi_e} \sin(\varphi_e t) \approx v_e t
\]

\[
y_e(t) = \frac{v_e}{\varphi_e} \left(1 - \cos(\varphi_e t)\right) \approx \frac{1}{2} v_e \varphi_e t^2
\]

\[
\beta_e(t) = \varphi_e t
\]
Examples for Motion Models: Lane Tracking

- Ground plane lane representation
  - Second or third order polynomial, e.g.

\[
 f(x, l) = \left( \frac{1}{6} c_1 l^3 \right) + \frac{1}{2} c_0 x^2 + \beta x + y_i
\]

\[
 x = [c_1 \quad c_0 \quad \beta \quad y_1 \quad \cdots \quad y_n]^T
\]

- Derived from the clothoid model
- Well known, simple model
- Does not cover all types of road geometry transitions

![2nd and 3rd order lane representations](image)
Examples for Motion Models: Lane Marking Tracking

- Driving along the parabola with $v_e t$ with a 3rd order model yields the following motion model

\[
x_{k|k-1} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
\frac{1}{2} (v_e t)^2 & v_e t & 1 & 0 & 0 \\
\frac{1}{6} (v_e t)^2 & \frac{1}{2} (v_e t)^2 & v_e t & 1 & 0 \\
\frac{1}{6} (v_e t)^2 & \frac{1}{2} (v_e t)^2 & v_e t & 0 & 1 \\
\end{bmatrix} x_{k-1|k-1} + \begin{bmatrix}
0 \\
0 \\
-t \phi_e \\
-\frac{1}{2} v_e t^2 \\
-\frac{1}{2} v_e t^2 \\
\end{bmatrix} w_k
\]
Examples for Motion Models: New Alternatives for Lane Tracking

- Spline description
- Set of control points
- More flexible geometry description
  - Adjustable control point density
  - Needs careful modelling

\[ f_1(x, l) \]
\[ f_2(x, l) \]
\[ f_3(x, l) \]
\[ f_4(x, l) \]
Examples for Motion Models: New Alternatives for Lane Tracking

• Spline tracker
  • The set of control points is included in the state vector of a Kalman filter
  • The control points are shifted according to the known ego-motion
  • New points are added on an extrapolated curve in front of the vehicle
  • Passed points are removed behind the vehicle
Lane Tracking: Design Selections

- Parallel lane markings in a single filter
  - More robust
  - Allows extraction of the pitch angle
  - Does not always match with reality

- Lane marking tracking using a bank of filters
  - Less robust
  - More flexible
  - Can model splitting and merging
Examples for Motion Models: Traffic Sign Tracking

- State is the position in 3D space
  \[ x = [x \ y \ z]^T \]
  - No target motion components
  - Ego motion compensation motion model
  - Size constraints can be used to refine the distance estimate
Examples for Motion Models: Pedestrian Tracking

- Ground plane motion
- Constant velocity target model
  - Relative $x$-velocity given as $v - v_e$
  - Yaw rate given as $-\varphi_e$
Examples for Motion Models: Pedestrian Tracking

- Ground-plane motion model
- CT model with Cartesian velocity and known turn rate (ego-motion)
- Size and ground-plane constraints can be used to give rough distance estimates

\[ x = [x \quad \dot{x} \quad y \quad \dot{y}]^T \]

No target rotation component, model can be found in Survey of Maneuvering Target Tracking.
Part I: Dynamic Models
X. Rong Li and Vesselin P. Jilkov
Examples for Motion Models: Vehicle Tracking

- CT Model with cartesian velocity
  - Ground plane motion model using rotating cartesian velocity vectors
  - Target turn rate is a part of the state vector
  - Relative CT Model with constant cartesian velocity
    - Rotation of the velocity components
  - Sometimes an extension with constant acceleration is used

- Propose to be used within an IMM estimator (with and without turn-rate) to improve filtering performance

\[ x = [x \ y \ \dot{x} \ \dot{y} \ \phi]^T \]
Examples for Motion Models: Vehicle Tracking

- CT Model with polar velocity
  - Ground plane motion model
  - Taylor-expanded at phi=0 to avoid a singularity
  - Unobservable at v=0
  - Sometimes an extension with constant acceleration is used

Observable dynamics and coordinate systems for vehicle tracking
Richard Altendorfer

- Better performance

- Used within an IMM estimator to avoid unobservability

\[ x = [x \ y \ \phi \ \dot{\phi} \ v]^T \]
Future Trends

• Robustness will remain a topic for image processing
• More exchange, interaction and fusion between several applications
• Real-time capable 3D reconstruction and ego motion extraction on embedded processors
• Vehicles will form a complex hereogenous sensor network
  • High delay communication with an environmental map over the internet
  • Low delay communication over C2C to nearby vehicles
  • Fusion with different sensors on the host vehicle (Camera, Radar, Gps, Map, Lidar, …)
  • Creation and update and upload of an environmental map
• Autonomous driving
Thank you.

Dr.-Ing. Mirko Meuter
Delphi Electronics & Safety
Delphiplatz 1, 42781 Wuppertal
02022912470
mirko.meuter@delphi.com