Beyond Kalman Filters

22.05.2014

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

• Single-Object Tracking
  - Background modeling
  - Template based tracking
  - Color based tracking
  - Contour based tracking
  - Tracking by online classification
  - Tracking-by-detection

• Bayesian Filtering
  - Kalman filters
  - Particle filters
  - Case studies

• Multi-Object Tracking

• Articulated Tracking
Today: Beyond Gaussian Error Models

Figure from Isard & Blake
Topics of This Lecture

• Recap: Kalman Filter
  - Basic ideas
  - Limitations
  - Extensions

• Particle Filters
  - Basic ideas
  - Propagation of general densities
  - Factored sampling

• Case study
  - Detector Confidence Particle Filter
  - Role of the different elements
Recap: Tracking as Inference

- **Inference problem**
  - The hidden state consists of the true parameters we care about, denoted $X$.
  - The measurement is our noisy observation that results from the underlying state, denoted $Y$.
  - At each time step, state changes (from $X_{t-1}$ to $X_t$) and we get a new observation $Y_t$.

- **Our goal: recover most likely state $X_t$ given**
  - All observations seen so far.
  - Knowledge about dynamics of state transitions.
Recap: Tracking as Induction

- **Base case:**
  - Assume we have initial prior that predicts state in absence of any evidence: $P(X_0)$
  - At the first frame, *correct* this given the value of $Y_0 = y_0$

- **Given corrected estimate for frame $t$:**
  - Predict for frame $t+1$
  - Correct for frame $t+1$
Recap: Prediction and Correction

- **Prediction:**
  
  \[ P(X_t | y_0, \ldots, y_{t-1}) = \int P(X_t | X_{t-1})P(X_{t-1} | y_0, \ldots, y_{t-1})dX_{t-1} \]

  - Dynamics model
  - Corrected estimate from previous step

- **Correction:**

  \[ P(X_t | y_0, \ldots, y_t) = \frac{P(y_t | X_t)P(X_t | y_0, \ldots, y_{t-1})}{\int P(y_t | X_t)P(X_t | y_0, \ldots, y_{t-1})dX_t} \]

  - Observation model
  - Predicted estimate

Slide credit: Svetlana Lazebnik
Recap: Linear Dynamic Models

- **Dynamics model**
  - State undergoes linear transformation $D_t$ plus Gaussian noise
  \[ x_t \sim N(D_t x_{t-1}, \Sigma_{d_t}) \]

- **Observation model**
  - Measurement is linearly transformed state plus Gaussian noise
  \[ y_t \sim N(M_t x_t, \Sigma_{m_t}) \]
Recap: Constant Velocity Model (1D)

- **State vector**: position $p$ and velocity $v$

\[
x_t = \begin{bmatrix} p_t \\ v_t \end{bmatrix}
\]

\[
p_t = p_{t-1} + (\Delta t)v_{t-1} + \varepsilon
\]

\[
v_t = v_{t-1} + \xi
\]

\[
x_t = D_t x_{t-1} + \text{noise} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_{t-1} \\ v_{t-1} \end{bmatrix} + \text{noise}
\]

- **Measurement is position only**

\[
y_t = M x_t + \text{noise} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} p_t \\ v_t \end{bmatrix} + \text{noise}
\]

(greek letters denote noise terms)
Recap: Constant Acceleration Model (1D)

- **State vector:** position $p$, velocity $v$, and acceleration $a$.

\[
x_t = \begin{bmatrix} p_t \\ v_t \\ a_t \end{bmatrix}
\]

\[
p_t = p_{t-1} + (\Delta t)v_{t-1} + \varepsilon
\]

\[
v_t = v_{t-1} + (\Delta t)a_{t-1} + \xi
\]

\[
a_t = a_{t-1} + \zeta
\]

\[
x_t = D_t x_{t-1} + \text{noise} = \begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_{t-1} \\ v_{t-1} \\ a_{t-1} \end{bmatrix} + \text{noise}
\]

- **Measurement is position only**

\[
y_t = Mx_t + \text{noise} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_t \\ v_t \\ a_t \end{bmatrix} + \text{noise}
\]

(greek letters denote noise terms)
Recap: General Motion Models

- Assuming we have differential equations for the motion
  - E.g. for (undamped) periodic motion of a pendulum
    \[ \frac{d^2 p}{dt^2} = -p \]

- Substitute variables to transform this into linear system
  \[
  \begin{align*}
p_1 &= p \\
p_2 &= \frac{dp}{dt} \\
p_3 &= \frac{d^2 p}{dt^2}
  \end{align*}
  \]

- Then we have
  \[
  \begin{align*}
x_t &= \begin{bmatrix} p_{1,t} \\ p_{2,t} \\ p_{3,t} \end{bmatrix} \\
p_{1,t} &= p_{1,t-1} + (\Delta t)p_{2,t-1} + \varepsilon \\
p_{2,t} &= p_{2,t-1} + (\Delta t)p_{3,t-1} + \xi \\
p_{3,t} &= -p_{1,t-1} + \zeta
  \end{align*}
  \]

  \[
  D_t = \begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & \Delta t \\ -1 & 0 & 0 \end{bmatrix}
  \]
Recap: The Kalman Filter

Know corrected state from previous time step, and all measurements up to the current one
→ Predict distribution over next state.

Receive measurement

Know prediction of state, and next measurement
→ Update distribution over current state.

Time update
(“Predict”)

Measurement update
(“Correct”)

Time advances: \( t++ \)

Mean and std. dev. of predicted state:

\[
\mu_t^-, \sigma_t^-
\]

Mean and std. dev. of corrected state:

\[
\mu_t^+, \sigma_t^+
\]

\[
P(X_t | y_0, \ldots, y_{t-1})
\]

\[
P(X_t | y_0, \ldots, y_t)
\]
Recap: General Kalman Filter (>1dim)

- What if state vectors have more than one dimension?

PREDICT

\[
x_t^- = D_t x_{t-1}^+
\]

\[
\Sigma_t^- = D_t \Sigma_{t-1}^+ D_t^T + \Sigma_d
\]

CORRECT

\[
K_t = \Sigma_t^- M_t^T \left( M_t \Sigma_t^- M_t^T + \Sigma_m \right)^{-1}
\]

\[
x_t^+ = x_t^- + K_t \left( y_t - M_t x_t^- \right)
\]

\[
\Sigma_t^+ = \left( I - K_t M_t \right) \Sigma_t^{-}
\]


- More weight on residual when measurement error covariance approaches 0.
- Less weight on residual as a priori estimate error covariance approaches 0.

for derivations, see F&P Chapter 17.3

Slide credit: Kristen Grauman
Resources: Kalman Filter Web Site

http://www.cs.unc.edu/~welch/kalman

- Electronic and printed references
  - Book lists and recommendations
  - Research papers
  - Links to other sites
  - Some software

- News

- Java-Based KF Learning Tool
  - On-line 1D simulation
  - Linear and non-linear
  - Variable dynamics
Remarks

- Try it!
  - Not too hard to understand or program

- Start simple
  - Experiment in 1D
  - Make your own filter in Matlab, etc.

- Note: the Kalman filter “wants to work”
  - Debugging can be difficult
  - Errors can go un-noticed

Slide adapted from Greg Welch
Topics of This Lecture

• Recap: Kalman Filter
  - Basic ideas
  - Limitations
  - Extensions

• Particle Filters
  - Basic ideas
  - Propagation of general densities
  - Factored sampling

• Case study
  - Detector Confidence Particle Filter
  - Role of the different elements
Extension: Extended Kalman Filter (EKF)

• Basic idea
  - State transition and observation model don’t need to be linear functions of the state, but just need to be differentiable.
    \[ x_t = f(x_{t-1}, u_t) + \epsilon \]
    \[ y_t = h(x_t) + \xi \]
  - The EKF essentially linearizes the nonlinearity around the current estimate by a Taylor expansion.

• Properties
  - Unlike the linear KF, the EKF is in general not an optimal estimator.
    - If the initial estimate is wrong, the filter may quickly diverge.
  - Still, it’s the de-facto standard in many applications
    - Including navigation systems and GPS
Kalman Filter - Other Extensions

- **Unscented Kalman Filter (UKF)**
  - Further development of EKF
  - Probability density is approximated by nonlinear transform of a random variable.
  - More accurate results than the EKF’s Taylor expansion approx.

- **Ensemble Kalman Filter (EnKF)**
  - Represents the distribution of the system state using a collection (an *ensemble*) of state vectors.
  - Replace covariance matrix by *sample covariance* from ensemble.
  - Still basic assumption that all prob. distributions involved are Gaussian.
  - EnKFs are especially suitable for problems with a large number of variables.
Even More Extensions

Switching linear dynamical system (SLDS):

\[ z_t \sim \pi_{z_{t-1}} \]
\[ x_t = A^{(z_t)} x_{t-1} + e_t(z_t) \]
\[ y_t = C x_t + w_t \]

\[ e_t \sim \mathcal{N}(0, \Sigma^{(z_t)}) \quad w_t \sim \mathcal{N}(0, R) \]

- **Switching Linear Dynamic System (SLDS)**
  - Use a set of \( k \) dynamic models \( A^{(1)}, \ldots, A^{(k)} \), each of which describes a different dynamic behavior.
  - Hidden variable \( z_t \) determines which model is active at time \( t \).
  - A switching process can change \( z_t \) according to distribution \( \pi_{z_{t-1}} \).

Figure source: Erik Sudderth
Topics of This Lecture

• Recap: Kalman Filter
  - Basic ideas
  - Limitations
  - Extensions

• Particle Filters
  - Basic ideas
  - Propagation of general densities
  - Factored sampling

• Case study
  - Detector Confidence Particle Filter
  - Role of the different elements

Today: only main ideas

Formal introduction next Tuesday
When Is A Single Hypothesis Too Limiting?

Initial position  Prediction  Measurement  Update

---

Figure from Thrun & Kosecka

Slide credit: Kristen Grauman
When Is A Single Hypothesis Too Limiting?

- Consider this example: say we are tracking the face on the right using a skin color blob to get our measurement.
When Is A Single Hypothesis Too Limiting?

- Consider this example: say we are tracking the face on the right using a skin color blob to get our measurement.

![Figure from Thrun & Kosecka](http://example.com/figure.png)

Video from Jojic & Frey

Slide credit: Kristen Grauman
Propagation of General Densities

Deterministic drift

Stochastic diffusion

Reactive effect of measurement

Slide credit: Svetlana Lazebnik

Figure from Isard & Blake
Factored Sampling

- **Idea:** Represent state distribution non-parametrically
  - **Prediction:** Sample points from prior density for the state, \( P(X) \)
  - **Correction:** Weight the samples according to \( P(Y | X) \)

\[
P(X_t | y_0, \ldots, y_t) = \frac{P(y_t | X_t)P(X_t | y_0, \ldots, y_{t-1})}{\int P(y_t | X_t)P(X_t | y_0, \ldots, y_{t-1})dX_t}
\]
Particle Filtering

• (Also known as Sequential Monte Carlo Methods)

• Idea
  ∘ We want to use sampling to propagate densities over time (i.e., across frames in a video sequence).
  ∘ At each time step, represent posterior $P(X_t | Y_t)$ with weighted sample set.
  ∘ Previous time step’s sample set $P(X_{t-1} | Y_{t-1})$ is passed to next time step as the effective prior.
Particle Filtering

Start with weighted samples from previous time step
Sample and shift according to dynamics model
Spread due to randomness; this is predicted density $P(X_t | Y_{t-1})$

Weight the samples according to observation density
Arrive at corrected density estimate $P(X_t | Y_t)$

M. Isard and A. Blake, **CONDENSATION -- conditional density propagation for visual tracking**, IJCV 29(1):5-28, 1998

Slide credit: Svetlana Lazebnik
Particle Filtering - Visualization

Code and video available from
http://www.robots.ox.ac.uk/~misard/condensation.html
Particle Filtering Results

http://www.robots.ox.ac.uk/~misard/condensation.html

Figure from Isard & Blake
Particle Filtering Results

- Some more examples

http://www.robots.ox.ac.uk/~misard/condensation.html

Videos from Isard & Blake
Obtaining a State Estimate

- Note that there’s no explicit state estimate maintained, just a “cloud” of particles.
- Can obtain an estimate at a particular time by querying the current particle set.
- Some approaches:
  - “Mean” particle
    - Weighted sum of particles
    - Confidence: inverse variance
  - Really want a mode finder—mean of tallest peak
Condensation: Estimating Target State

State samples (thickness proportional to weight)  
Mean of weighted state samples

From Isard & Blake, 1998
Summary: Particle Filtering

• **Pros:**
  - Able to represent arbitrary densities
  - Converging to true posterior even for non-Gaussian and nonlinear system
  - Efficient: particles tend to focus on regions with high probability
  - Works with many different state spaces
    - E.g. articulated tracking in complicated joint angle spaces
  - Many extensions available
Summary: Particle Filtering

• **Cons / Caveats:**
  - #Particles is important performance factor
    - Want as few particles as possible for efficiency.
    - But need to cover state space sufficiently well.
  - Worst-case complexity grows exponentially in the dimensions
  - Multimodal densities possible, but still single object
    - Interactions between multiple objects require special treatment.
    - Not handled well in the particle filtering framework (state space explosion).
Topics of This Lecture

- Recap: Kalman Filter
  - Basic ideas
  - Limitations
  - Extensions

- Particle Filters
  - Basic ideas
  - Propagation of general densities
  - Factored sampling

- Case study
  - Detector Confidence Particle Filter
  - Role of the different elements
Challenge: Unreliable Object Detectors

- Example:
  - Low-res webcam footage (320×240), MPEG compressed

Detector input  Tracker output

How to get from here...  ?  ...to here?
Tracking based on Detector Confidence

- Detector output is often not perfect
  - Missing detections and false positives
  - But continuous confidence still contains useful cues.

- Idea employed here:
  - Use continuous detector confidence to track persons over time.
Main Ideas

- Detector confidence particle filter
  - Initialize particle cloud on strong object detections.
  - Propagate particles using continuous detector confidence as observation model.

- Disambiguate between different persons
  - Train a person-specific classifier with online boosting.
  - Use classifier output to distinguish between nearby persons.

[Breitenstein, Reichlin, Leibe et al., ICCV’09]
Detector Confidence Particle Filter

- **State:** \( x = \{x, y, u, v\} \)

- **Motion model (constant velocity)**
  \[
  \begin{align*}
  (x, y)_t &= (x, y)_{t-1} + (u, v)_{t-1} \cdot \Delta t + \varepsilon(x, y) \\
  (u, v)_t &= (u, v)_{t-1} + \varepsilon(u, v)
  \end{align*}
  \]

- **Observation model**
  \[
  w_{tr,p} = p(y_t | x_t^{(i)}) = \beta \cdot I(tr) \cdot p_N(p - d^*) + \gamma \cdot d_c(p) \cdot p_o(tr) + \eta \cdot c_{tr}(p)
  \]

Discrete detections  Detector confidence  Classifier confidence
When Is Which Term Useful?

- Discrete detections
- Detector confidence
- Classifier confidence
Each Observation Term Increases Robustness!

<table>
<thead>
<tr>
<th>Observation Model Terms</th>
<th>MOTP</th>
<th>MOTA</th>
<th>FN</th>
<th>FP</th>
<th>ID Sw.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Det+DetConf+Class</td>
<td>70.0%</td>
<td>72.9%</td>
<td>26.8%</td>
<td>0.3%</td>
<td>0</td>
</tr>
<tr>
<td>2: Det+DetConf</td>
<td>64.0%</td>
<td>54.5%</td>
<td>28.2%</td>
<td>17.2%</td>
<td>5</td>
</tr>
<tr>
<td>3: Det+Class</td>
<td>65.0%</td>
<td>55.3%</td>
<td>31.3%</td>
<td>13.4%</td>
<td>0</td>
</tr>
<tr>
<td>4: Det</td>
<td>67.0%</td>
<td>40.9%</td>
<td>30.7%</td>
<td>28.0%</td>
<td>10</td>
</tr>
</tbody>
</table>

Detector only

CLEAR MOT scores
Each Observation Term Increases Robustness!

<table>
<thead>
<tr>
<th>Observation Model Terms</th>
<th>MOTP</th>
<th>MOTA</th>
<th>FN</th>
<th>FP</th>
<th>ID Sw.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Det + DetConf + Class</td>
<td>70.0%</td>
<td>72.9%</td>
<td>26.8%</td>
<td>0.3%</td>
<td>0</td>
</tr>
<tr>
<td>2: Det + DetConf</td>
<td>64.0%</td>
<td>54.5%</td>
<td>28.2%</td>
<td>17.2%</td>
<td>5</td>
</tr>
<tr>
<td>3: Det + Class</td>
<td>65.0%</td>
<td>55.3%</td>
<td>31.3%</td>
<td>13.4%</td>
<td>0</td>
</tr>
<tr>
<td>4: Det</td>
<td>67.0%</td>
<td>40.9%</td>
<td>30.7%</td>
<td>28.0%</td>
<td>10</td>
</tr>
</tbody>
</table>

Detector + Confidence

CLEAR MOT scores

B. Leibe
Each Observation Term Increases Robustness!

<table>
<thead>
<tr>
<th>Observation Model Terms</th>
<th>MOTP</th>
<th>MOTA</th>
<th>FN</th>
<th>FP</th>
<th>ID Sw.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Det+DetConf+Class</td>
<td>70.0%</td>
<td>72.9%</td>
<td>26.8%</td>
<td>0.3%</td>
<td>0</td>
</tr>
<tr>
<td>2: Det+DetConf</td>
<td>64.0%</td>
<td>54.5%</td>
<td>28.2%</td>
<td>17.2%</td>
<td>5</td>
</tr>
<tr>
<td>3: Det+Class</td>
<td>65.0%</td>
<td>55.3%</td>
<td>31.3%</td>
<td>13.4%</td>
<td>0</td>
</tr>
<tr>
<td>4: Det</td>
<td>67.0%</td>
<td>40.9%</td>
<td>30.7%</td>
<td>28.0%</td>
<td>10</td>
</tr>
</tbody>
</table>

Detector + Classifier

CLEAR MOT scores

B. Leibe
Each Observation Term Increases Robustness!

<table>
<thead>
<tr>
<th>Observation Model Terms</th>
<th>MOTP</th>
<th>MOTA</th>
<th>FN</th>
<th>FP</th>
<th>ID Sw.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Det+DetConf+Class</td>
<td>70.0%</td>
<td>72.9%</td>
<td>26.8%</td>
<td>0.3%</td>
<td>0</td>
</tr>
<tr>
<td>2: Det+DetConf</td>
<td>64.0%</td>
<td>54.5%</td>
<td>28.2%</td>
<td>17.2%</td>
<td>5</td>
</tr>
<tr>
<td>3: Det+Class</td>
<td>65.0%</td>
<td>55.3%</td>
<td>31.3%</td>
<td>13.4%</td>
<td>0</td>
</tr>
<tr>
<td>4: Det</td>
<td>67.0%</td>
<td>40.9%</td>
<td>30.7%</td>
<td>28.0%</td>
<td>10</td>
</tr>
</tbody>
</table>

Detector + Confidence + Classifier

False negatives, false positives, and ID switches decrease!

CLEAR MOT scores
Qualitative Results
Remaining Issues

- Some false positive initializations at wrong scales...
  - Due to limited scale range of the person detector.
  - Due to boundary effects of the person detector.
References and Further Reading

• A good tutorial on Particle Filters

• The CONDENSATION paper