CS231A
Computer Vision: From 3D Reconstruction to Recognition
Optimal Estimation
Perception as a Continuous Process
Perception as a Multi-Modal Experience
Perception as Inference
Recursive State Estimation

Mathematical Formalism to:

○ continuously integrate measurements
○ from different sensor sources
○ to infer the state of a latent variable
What is a state? What is a representation?

Hidden Markov Model
Representations for Autonomous Driving

x: pose, size, type
z: Lidar, Stereo or RGB

Image adapted from NuScenes by Motional. nuscenes.org
Representations for Manipulation

x: 6 DoF Object Pose, whether pixels are occluded
z: Dense Depth Images

Manuel Wühtrich et al. "Probabilistic Object Tracking using a Depth Camera", IROS 2013
Why do we care about state estimation in Robotics?

**Partially Observable Markov Decision Process**

- **Control Input**: \( u(t) \), \( u(t+1) \), \( u(t+2) \)
- **State**: \( x(t-1) \), \( x(t) \), \( x(t+1) \)
- **Observation**: \( z(t-1) \), \( z(t) \), \( z(t+1) \)
Today

- Intro: Why state estimation?
- Bayes Filter
- Kalman Filter
- Extended Kalman Filter

For more depth:
- AA 273: State Estimation and Filtering for Robotic Perception – Mac Schwager
The Agent and the Environment

\( x_t \) \quad \text{Decision Making} \quad \text{State} \quad \text{Perception} \quad \text{(Environment) state constantly changes} \quad \text{\( u_t \)} 

\( z_{t+1} \)
Notation

\[ x(t-1) \rightarrow x(t) \rightarrow x(t+1) \]

\[ z(t-1) \rightarrow z(t) \rightarrow z(t+1) \]

\[ x \quad \text{State of dynamical system, dim n} \]
\[ x_t \quad \text{Instantiation of system state at time } t \]
\[ z \quad \text{Sensor Observation Vector, dim k} \]
\[ z_t \quad \text{Specific Observation at time } t \]
\[ u \quad \text{Robot action / control input, dim m} \]
\[ u_t \quad \text{Robot action / control input at time } t \]
\[ p(x_t | z_{0:t}, u_{0:t}) \quad \text{Probability distribution} \]

Markov Assumption
State is complete
Probabilistic Generative Laws

• Evolution of state and measurement governed by probabilistic laws

• $x_t$ generated stochastically
State Transition Model

• Probability distribution conditioned on all previous states, measurements and controls

\[ p(x_t \mid x_{0:t-1}, z_{1:t-1}, u_{1:t}) \]

• Assumption: State complete

\[ p(x_t \mid x_{0:t-1}, z_{1:t-1}, u_{1:t}) = p(x_t \mid x_{t-1}, u_t) \]
Measurement Model

- Probability distribution conditioned on all previous states, measurements and controls

\[ p(z_t \mid x_{0:t}, z_{1:t-1}, u_{1:t}) \]

- Assumption: State complete

\[ p(z_t \mid x_{0:t}, z_{1:t-1}, u_{1:t}) = p(z_t \mid x_t) \]
Belief Distribution

- Assigns probability to each possible hypothesis about what the true state may be
- Posterior distributions over state conditioned on all the data

\[
\text{bel}(x_t) = p(x_t | z_{1:t}, u_{1:t})
\]

- Before incorporating measurement \( Z_t = \) prediction

\[
\overline{\text{bel}}(x_t) = p(x_t | z_{1:t-1}, u_{1:t})
\]
The Bayes Filter

• Recursive filter for estimating $x_t$ only from $x_{t-1}, z_t$ and $u_t$ and not from the ever-growing history $z_{1:t}, u_{1:t}$

```
1: Algorithm Bayes filter($bel(x_{t-1}), u_t, z_t$):
2:   for all $x_t$ do
3:     \[ \overline{bel}(x_t) = \int p(x_t | u_t, x_{t-1}) \, bel(x_{t-1}) \, dx \] 
4:     \[ bel(x_t) = \eta \, p(z_t | x_t) \, bel(x_t) \] 
5:   endfor
6: return $bel(x_t)$
```
Simple example – Belief & Measurement Model

Figure 2.2 A mobile robot estimating the state of a door.

\[
\begin{align*}
\text{bel}(X_0 = \text{open}) &= 0.5 \\
\text{bel}(X_0 = \text{closed}) &= 0.5 \\
p(Z_t = \text{sense\_open} | X_t = \text{is\_open}) &= 0.6 \\
p(Z_t = \text{sense\_closed} | X_t = \text{is\_open}) &= 0.4 \\
p(Z_t = \text{sense\_open} | X_t = \text{is\_closed}) &= 0.2 \\
p(Z_t = \text{sense\_closed} | X_t = \text{is\_closed}) &= 0.8
\end{align*}
\]
Simple example – Transition Model

Figure 2.2 A mobile robot estimating the state of a door.

\[
\begin{align*}
  p(X_t = \text{is.open} \mid U_t = \text{push}, X_{t-1} = \text{is.open}) &= 1 \\
  p(X_t = \text{is.closed} \mid U_t = \text{push}, X_{t-1} = \text{is.open}) &= 0 \\
  p(X_t = \text{is.open} \mid U_t = \text{push}, X_{t-1} = \text{is.closed}) &= 0.8 \\
  p(X_t = \text{is.closed} \mid U_t = \text{push}, X_{t-1} = \text{is.closed}) &= 0.2 \\
  p(X_t = \text{is.open} \mid U_t = \text{do.nothing}, X_{t-1} = \text{is.open}) &= 1 \\
  p(X_t = \text{is.closed} \mid U_t = \text{do.nothing}, X_{t-1} = \text{is.open}) &= 0 \\
  p(X_t = \text{is.open} \mid U_t = \text{do.nothing}, X_{t-1} = \text{is.closed}) &= 0 \\
  p(X_t = \text{is.closed} \mid U_t = \text{do.nothing}, X_{t-1} = \text{is.closed}) &= 1
\end{align*}
\]
The Bayes Filter - Derivation

• Bayes Rule

\[ p(a, b) = p(a \mid b)p(b) = p(b \mid a)p(a) \]

\[ p(a \mid b) = \frac{p(b \mid a)p(a)}{p(b)} \]

\[ p(x_t \mid z_{1:t}, u_{1:t}) = \frac{p(z_t \mid x_t, z_{1:t-1}, u_{1:t}) p(x_t \mid z_{1:t-1}, u_{1:t})}{p(z_t \mid z_{1:t-1}, u_{1:t})} \] Normalization
The Bayes Filter - Derivation

- State is complete

\[ p(z_t \mid x_t, z_{1:t-1}, u_{1:t}) = p(z_t \mid x_t) \]

- Simplify

\[
p(x_t \mid z_{1:t}, u_{1:t}) = \frac{p(z_t \mid x_t, z_{1:t-1}, u_{1:t}) \ p(x_t \mid z_{1:t-1}, u_{1:t})}{p(z_t \mid z_{1:t-1}, u_{1:t})} = \eta \ p(z_t \mid x_t, z_{1:t-1}, u_{1:t}) \ p(x_t \mid z_{1:t-1}, u_{1:t})
\]

\[ = \eta \ p(z_t \mid x_t) \ p(x_t \mid z_{1:t-1}, u_{1:t}) \]

simplified
The Bayes Filter - Derivation

\[ p(x_t \mid z_{1:t}, u_{1:t}) = \eta p(z_t \mid x_t) p(x_t \mid z_{1:t-1}, u_{1:t}) \]

\[ bel(x_t) = \eta p(z_t \mid x_t) \overline{bel}(x_t) \]

This still depends on entire history

---

1: \textbf{Algorithm Bayes filter} \( \overline{bel}(x_{t-1}), u_t, z_t \):
2: \hspace{1em} for all \( x_t \) do
3: \hspace{2em} \overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1}) \overline{bel}(x_{t-1}) \, dx \]
4: \hspace{2em} \textcolor{green}{bel}(x_t) = \eta p(z_t \mid x_t) \overline{bel}(x_t) \hspace{1em} \text{Update Step}
5: \hspace{1em} \text{endfor}
6: \hspace{1em} \text{return} \overline{bel}(x_t) \hspace{1em} \text{Measurement Model}
The Bayes Filter - Derivation

\[ \text{bel}(x_t) = \eta p(z_t | x_t) \overline{\text{bel}}(x_t) \]

- Total probability  \( p(a) = \int p(a|b)p(b)db \)

\[ \overline{\text{bel}}(x_t) = p(x_t | z_{1:t-1}, u_{1:t}) \]
\[ = \int p(x_t | x_{t-1}, z_{1:t-1}, u_{1:t}) p(x_{t-1} | z_{1:t-1}, u_{1:t}) dx_{t-1} \]

- State is complete

\[ p(x_t | x_{t-1}, z_{1:t-1}, u_{1:t}) = p(x_t | x_{t-1}, u_t) \]
The Bayes Filter - Derivation

\[ p(x_t \mid x_{t-1}, z_{1:t-1}, u_{1:t}) = p(x_t \mid x_{t-1}, u_t) \]

\[
\text{bel}(x_t) = p(x_t \mid z_{1:t-1}, u_{1:t}) \\
= \int p(x_t \mid x_{t-1}, z_{1:t-1}, u_{1:t}) p(x_{t-1} \mid z_{1:t-1}, u_{1:t}) \, dx_{t-1} \\
= \int \underbrace{p(x_t \mid x_{t-1}, u_t)} \underbrace{p(x_{t-1} \mid z_{1:t-1}, u_{1:t-1})} \, dx_{t-1}
\]

Algorithm Bayes filter (bel(x_{t-1}), u_t, z_t):

1: for all \( x_t \) do
2: \( \text{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1}) \text{bel}(x_{t-1}) \, dx \) \hspace{1cm} \text{Predict Step}
3: \( \text{bel}(x_t) = \eta p(z_t \mid x_t) \text{bel}(x_t) \)
4: endfor
5: return \( \text{bel}(x_t) \)
Limitations

1. $p(x)$ is defined $\forall x$ – intractable
   - Discrete and small spaces
   - Continuous and/or large spaces – Moments, Finite # of samples

2. The integral term -> costly to compute
The Bayes Filter

- Recursive filter for estimating $x_t$ only from $x_{t-1}$, $z_t$ and $u_t$ and not from the ever-growing history $z_{1:t}, u_{1:t}$

```
1: Algorithm Bayes filter(bel(x_{t-1}), u_t, z_t):
2:     for all x_t do
3:         bel(x_t) = \int p(x_t | u_t, x_{t-1}) bel(x_{t-1}) dx
4:         bel(x_t) = \eta p(z_t | x_t) bel(x_t)
5:     endfor
6: return bel(x_t)
```

Transition/Dynamics model

Predict Step

Update Step

Measurement Model
Gaussian Filters - Kalman Filter

\[ x \sim N(\mu, \Sigma) \]

\[ p(x) = \text{det}(2\pi \Sigma)^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu) \right\} \]
Kalman Filter

- Gaussian Belief
- Linear Transition Model
  \[ x_t = A_t x_{t-1} + B_t u_t + \epsilon_t \]
  \[ x_t = \begin{pmatrix} x_{1,t} \\ x_{2,t} \\ \vdots \\ x_{n,t} \end{pmatrix} \]
  \[ \epsilon_t \sim N(0, R) \]
  \[ u_t = \begin{pmatrix} u_{1,t} \\ u_{2,t} \\ \vdots \\ u_{m,t} \end{pmatrix} \]

- Linear Measurement Model
  \[ z_t = C_t x_t + \delta_t \]
  \[ \delta_t \sim N(0, Q) \]
Kalman Filter

- **Initial Belief** \( x_0 \sim N(\mu_0, \Sigma_0) \)

\[
bel(x_0) = p(x_0) = \det(2\pi\Sigma_0)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2} (x_0 - \mu_0)^T \Sigma_0^{-1} (x_0 - \mu_0)\right\}
\]

- **Distribution over next state**

\[
p(x_t | u_t, x_{t-1}) = \det(2\pi R_t)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2} \left( x_t - A_t x_{t-1} - B_t u_t \right)^T R_t^{-1} \left( x_t - A_t x_{t-1} - B_t u_t \right)\right\}
\]

- **Likelihood of Measurement**

\[
p(z_t | x_t) = \det(2\pi Q_t)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2} (z_t - C_t x_t)^T Q_t^{-1} (z_t - C_t x_t)\right\}
\]
# The Kalman Filter Algorithm

1: \textbf{Algorithm Kalman filter}(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t):

2: \quad \bar{\mu}_t = A_t \mu_{t-1} + B_t u_t

3: \quad \bar{\Sigma}_t = A_t \Sigma_{t-1} A^T_t + R_t

4: \quad K_t = \bar{\Sigma}_t C^T_t (C_t \bar{\Sigma}_t C^T_t + Q_t)^{-1}

5: \quad \mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)

6: \quad \Sigma_t = (I - K_t C_t) \bar{\Sigma}_t

7: \quad \text{return } \mu_t, \Sigma_t

Uncertainty increases

\[ K = \text{Kalman Gain} \quad K \approx \frac{R}{Q} \]

Uncertainty decreases

If \( R \) large, then \( K \) is large. Update dominated by innovation.

If \( Q \) large, then \( K \) is small. Update dominated by prediction.

---

1: \textbf{Algorithm Bayes filter}(bel(x_{t-1}), u_t, z_t):

2: \quad \text{for all } x_t \text{ do}

3: \quad \text{Predict Step } bel(x_t) = \int p(x_t | u_t, x_{t-1}) \text{ bel}(x_{t-1}) \, dx

4: \quad \text{Update Step } bel(x_t) = \eta p(z_t | x_t) \text{ bel}(x_t)

5: \quad \text{endfor}

6: \quad \text{return } bel(x_t)
Example

\[ p(x_0) \]

Measurement

\[ bel(x_0) \]

After Update

\[ p(z_0 | x_0) \]

Measurement

\[ bel(x_1) \]

After Prediction

\[ p(z_1 | x_1) \]

Measurement

\[ bel(x_1) \]

After Update
Propagating a Gaussian through a Linear Model

\[ y = ax + b \]

Mean \( \mu \)
Propagating a Gaussian through a Non-Linear Model
Linearizing the Non-Linear Model

![Diagram showing linearization of a non-linear model]
Representations for Manipulation

Manuel Wühtrich et al. “Probabilistic Object Tracking using a Depth Camera”, IROS 2013
Extended Kalman filter - Process Model

\[ x_t = g(u_t, x_{t-1}) + \varepsilon_t \quad \text{Process Model} \]
\[ z_t = h(x_t) + \delta_t. \quad \text{Measurement Model} \]

First order Taylor Expansion – linear approximation around value and slope

\[
g'(u_t, x_{t-1}) := \frac{\partial g(u_t, x_{t-1})}{\partial x_{t-1}} \quad \text{Gradient of Nonlinear function around } x_{t-1}
\]

\[
g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + g'(u_t, \mu_{t-1}) (x_{t-1} - \mu_{t-1}) =: G_t
\]
\[
= g(u_t, \mu_{t-1}) + G_t (x_{t-1} - \mu_{t-1}) \quad \text{Jacobian}
\]
Extended Kalman filter -
Process Model

\[
g(u_t, x_{t-1}) \approx \underbrace{g(u_t, \mu_{t-1}) + g'(u_t, \mu_{t-1})(x_{t-1} - \mu_{t-1})}_{=: G_t}
= g(u_t, \mu_{t-1}) + G_t(x_{t-1} - \mu_{t-1})
\]

Same equations as in previous slide

Written as Gaussian:

\[
p(x_t \mid u_t, x_{t-1}) 
\approx \det(2\pi R_t)^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} \left[ x_t - g(u_t, \mu_{t-1}) - G_t (x_{t-1} - \mu_{t-1}) \right]^T R_t^{-1} \left[ x_t - g(u_t, \mu_{t-1}) - G_t (x_{t-1} - \mu_{t-1}) \right] \right\}
\]
Extended Kalman Filter – Measurement Model

\[ x_t = g(u_t, x_{t-1}) + \varepsilon_t \quad \text{Process Model} \]

\[ z_t = h(x_t) + \delta_t. \quad \text{Measurement Model} \]

First order Taylor Expansion – linear approximation around value and slope

\[
\begin{align*}
    h(x_t) & \approx h(\bar{\mu}_t) + h'(\bar{\mu}_t)(x_t - \bar{\mu}_t) \\
         &= H_t (x_t - \bar{\mu}_t)
\end{align*}
\]

\[ = h(\bar{\mu}_t) + H_t (x_t - \bar{\mu}_t) \quad \text{Jacobian} \]

Written as Gaussian:

\[
p(z_t \mid x_t) = \det (2\pi Q_t)^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} [z_t - h(\bar{\mu}_t) - H_t (x_t - \bar{\mu}_t)]^T Q_t^{-1} [z_t - h(\bar{\mu}_t) - H_t (x_t - \bar{\mu}_t)] \right\}
\]
The Extended Kalman Filter Algorithm

1: \textbf{Algorithm Extended Kalman filter}(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t):

2: \quad \bar{\mu}_t = g(u_t, \mu_{t-1}) \quad \text{Predict}

3: \quad \tilde{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t

4: \quad K_t = \tilde{\Sigma}_t H_t^T (H_t \tilde{\Sigma}_t H_t^T + Q_t)^{-1} \quad \text{Update}

5: \quad \mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))

6: \quad \Sigma_t = (I - K_t H_t) \tilde{\Sigma}_t

7: \quad \text{return } \mu_t, \Sigma_t

---

<table>
<thead>
<tr>
<th>state prediction (Line 2)</th>
<th>Kalman filter</th>
<th>EKF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$A_t \mu_{t-1} + B_t u_t$</td>
<td>$g(u_t, \mu_{t-1})$</td>
</tr>
<tr>
<td>measurement prediction (Line 5)</td>
<td>$C_t \bar{\mu}_t$</td>
<td>$h(\bar{\mu}_t)$</td>
</tr>
</tbody>
</table>
CS231
Introduction to Computer Vision

Next lecture:
Optimal Estimation cont’