Localization

Howie Choset
Adapted from slides by Humphrey Hu, Trevor Decker, and Brad Neuman
Localization

• General robotic task
  – “Where am I?”

• Techniques generalize to many estimation tasks
  – System parameter estimation
  – Noisy signal smoothing
  – Weather system modeling
Localization Problem Definition

• Goal: Estimate state given a history of observations and actions
  – State: Information sufficient to predict observations
  – Observation: Information derived from state
  – Actions: Inputs that affect the state

• State is important, but what is it?
State

- Any parameterization to describe our system

- Examples
  - Car / Planar Robot
    - Minimal \((x, y, \Theta)\)
    - Could be \((x, y, \Theta, \text{temperature, time of day, Google stock, favorite color, etc.})\)
  - Consider slot car
    - Only 1D problem!
Action / Control Input

- “Things we can do”
- Can be discrete set:
  - Pacman left, right, up, down
- Or continuous inputs:
  - Helicopter throttles
Observation

• Information derive from a sensor

• Examples
  – Time (from a clock)
  – Temperature (thermoater)
  – Encoder

• Good vs. Bad (really strong vs. weak correlations)
  – ie. Temperature at a city for estimating a weather system
Absolute vs. Relative

• Absolute Localization
  – Defines state relative to a common (usually fixed) reference frame
  – Useful for coordination, navigation, etc.

• Relative Localization
  – Defines state relative to a local (non-shared) reference frame
  – Useful for exploration, displacement estimation
Why not GPS/Vicon?

- Signal-denied environments
- Insufficient performance
  - Accuracy
  - Bandwidth (update rate)
  - Bias
  - Receiver size
Why not odometry?

• Uncertainty and error accumulation!
  – Unmodeled environmental factors
  – Integration errors
  – Modeling errors

• Bottom line:

  *Uncertainty is a part of life.*
  *We have to deal with it!*
Localization Problem Definition

• Goal: Estimate state given a history of noisy observations and noisy actions
  – **State**: Information sufficient to predict observations
  – **Observation**: Information derived from state
  – **Actions**: Inputs that affect the state
Localization: Estimate State

- Move: Motion Model
- Observe: Observation Model
Example

- Moving only in one dimension
- Known map of flower garden
- Simple flower detector
  - Beeps when you are in front of a flower
  - Gaussian distribution of a flower given a beep
Initially, no idea where we are
Example

- First observation update
Example

- New belief about location
Example

- Robot moves, motion update
Example

- Observation update
Example

Final Belief
Differential Drive Motion Model

- **State:** \((x, y, \Theta)\) \(SE2\) pose
- **Actions:**
  - Drive forward, angle --OR--
  - Drive wheel 1, drive wheel 2
- **Measurements:** Wheel rotation ticks

2-wheeled Lego Robot
Differential Drive Example

- Run the same trajectory many times
  - They’re all different!
  - Why?

Trial 1
Trial 2
Trial 3
... Trial n
Differential Drive Example

• Run a 10 cm straight trajectory many times
• Look at the results as a distribution

# of occurrences

Displacement $d$

0 10 20
Differential Dive Example

- Run a 10 cm straight trajectory many times
- Look at the results as a distribution

\[ \eta \sim \mathcal{N}(\mu, \sigma^2) \]

\[ x^{t+1} = f(x^t) + \eta \]
Differential Drive Example

- Subtract out model contribution to determine noise component

\[ x^{t+1} = f(x^t, u^t) + \eta^t \]
Odometry Example

- Differential drive robot experiences uncertainty in distance traveled and heading
  - Produces a “banana” distribution
  - Hard to model!
Differential Drive Sensor Model

• Recall our odometry equations:

\[
\begin{align*}
v_l &= \frac{\text{Current Encoder Ticks (left motor)} - \text{Encoder Ticks saved from previous loop (left motor)}}{\text{Time elapsed since we last polled the encoders}} \\
v_r &= \frac{\text{Current Encoder Ticks (right motor)} - \text{Encoder Ticks saved from previous loop (right motor)}}{\text{Time elapsed since we last polled the encoders}}
\end{align*}
\]

\[
V_l = v_l R \\
V_r = v_r R \\
v = \frac{V_r + V_l}{2} \\
\omega = \frac{V_r - V_l}{L}
\]

\[
\begin{pmatrix}
\dot{x} \\
\dot{y} \\
\dot{\theta}
\end{pmatrix} = \begin{pmatrix}
\nu \cos(\theta) \\
\nu \sin(\theta) \\
\omega
\end{pmatrix}
\]

\[
k_{00} = \nu \cos(\theta_{n-1}) \\
k_{01} = \nu \sin(\theta_{n-1}) \\
k_{02} = \omega \\
k_{10} = \nu \cos(\theta_{n-1} + \frac{1}{2} k_{02}) \\
k_{11} = \nu \sin(\theta_{n-1} + \frac{1}{2} k_{02}) \\
k_{12} = \omega \\
k_{20} = \nu \cos(\theta_{n-1} + \frac{1}{2} k_{12}) \\
k_{21} = \nu \sin(\theta_{n-1} + \frac{1}{2} k_{12}) \\
k_{22} = \omega \\
k_{30} = \nu \cos(\theta_{n-1} + t k_{22}) \\
k_{31} = \nu \sin(\theta_{n-1} + t k_{22}) \\
k_{32} = \omega
\]

\[
\vec{x}_n = \begin{pmatrix}
x_n \\
y_n \\
\theta_n
\end{pmatrix} = \binom{x_{n-1}}{y_{n-1}} + \frac{1}{6} \begin{pmatrix}
k_{00} + 2 (k_{10} + k_{20}) + k_{30} \\
k_{01} + 2 (k_{11} + k_{21}) + k_{31} \\
k_{02} + 2 (k_{12} + k_{22}) + k_{32}
\end{pmatrix}
\]

"initial" state
Differential Drive Sensor Model

- Recall our odometry equations:

\[
\begin{align*}
    v_l &= \frac{\text{Current EncoderTicks (left motor)} - \text{EncoderTicks saved from previous loop (left motor)}}{\text{Time elapsed since we last polled the encoders}} \\
    v_r &= \frac{\text{Current EncoderTicks (right motor)} - \text{EncoderTicks saved from previous loop (right motor)}}{\text{Time elapsed since we last polled the encoders}}
\end{align*}
\]

\[
\begin{align*}
    V_l &= v_l R \\
    V_r &= v_r R \\
    v &= \frac{V_r + V_l}{2} \\
    \omega &= \frac{V_r - V_l}{L}
\end{align*}
\]

\[
\begin{pmatrix}
    \dot{x} \\
    \dot{y} \\
    \dot{\theta}
\end{pmatrix} = \begin{pmatrix}
    v \cos(\theta) \\
    v \sin(\theta) \\
    \omega
\end{pmatrix}
\]

\[
\begin{align*}
    k_{00} &= v \cos(\theta_{n-1}) \\
    k_{01} &= v \sin(\theta_{n-1}) \\
    k_{02} &= \omega \\
    k_{10} &= v \cos(\theta_{n-1} + \frac{1}{2} \ k_{02}) \\
    k_{11} &= v \sin(\theta_{n-1} + \frac{1}{2} \ k_{02}) \\
    k_{12} &= \omega \\
    k_{20} &= v \cos(\theta_{n-1} + \frac{3}{2} \ k_{02}) \\
    k_{21} &= v \sin(\theta_{n-1} + \frac{3}{2} \ k_{02}) \\
    k_{22} &= \omega \\
    k_{30} &= v \cos(\theta_{n-1} + t \ k_{22}) \\
    k_{31} &= v \sin(\theta_{n-1} + t \ k_{22}) \\
    k_{32} &= \omega
\end{align*}
\]

Motion

"initial" state
A Brief Overview of

PROBABILITY
Discrete Probability Distribution

- Let $X$ be the value of a die roll
- $X$ is unknown (a Random Variable)
- $P(X = v)$ means “Probability that we sample $X$ and it equals $v$”

<table>
<thead>
<tr>
<th>$v$</th>
<th>$P(X=v)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1/6</td>
</tr>
<tr>
<td>2</td>
<td>1/6</td>
</tr>
<tr>
<td>3</td>
<td>1/6</td>
</tr>
<tr>
<td>4</td>
<td>1/6</td>
</tr>
<tr>
<td>5</td>
<td>1/6</td>
</tr>
<tr>
<td>6</td>
<td>1/6</td>
</tr>
</tbody>
</table>
Discrete Probability Distribution

- Let $X$ be the value of a die roll
- $X$ is unknown (a Random Variable)
- $P(X = v)$ means “Probability that we sample $X$ and it equals $v$”

<table>
<thead>
<tr>
<th>$v$</th>
<th>$P(X=v)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1/6</td>
</tr>
<tr>
<td>2</td>
<td>1/6</td>
</tr>
<tr>
<td>3</td>
<td>1/6</td>
</tr>
<tr>
<td>4</td>
<td>1/6</td>
</tr>
<tr>
<td>5</td>
<td>1/6</td>
</tr>
<tr>
<td>6</td>
<td>1/6</td>
</tr>
</tbody>
</table>

Sums to 1
Discrete Probability Distribution

- This time, $X$ is a weighted die
- This is a different distribution for the same variable

<table>
<thead>
<tr>
<th>$v$</th>
<th>$P(X=v)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>2</td>
<td>0.1</td>
</tr>
<tr>
<td>3</td>
<td>0.1</td>
</tr>
<tr>
<td>4</td>
<td>0.2</td>
</tr>
<tr>
<td>5</td>
<td>0.25</td>
</tr>
<tr>
<td>6</td>
<td>0.25</td>
</tr>
</tbody>
</table>
Discrete Probability Distribution

- Consider a sum of dice

\[
P(X_1 = v) = \frac{1}{6}
\]

\[
P(X_2 = v) = \frac{1}{6}
\]

\[
\begin{align*}
V & \quad P(X_1 = v) & \quad P(X_1 + X_2 = v) \\
1 & \quad \frac{1}{6} & \quad P(X_1 = 1) \times P(X_2 = 1) \\
2 & \quad \frac{1}{6} & \quad P(X_1 = 1) \times P(X_2 = 2) + P(X_1 = 2) \times P(X_2 = 1) \\
3 & \quad \frac{1}{6} & \quad \ldots \\
4 & \quad \frac{1}{6} & \quad P(X_1 = 1) \times P(X_2 = 3) + \ldots \\
5 & \quad \frac{1}{6} & \quad \ldots \\
6 & \quad \frac{1}{6} & \quad \ldots \\
7 & \quad \frac{1}{6} & \quad \ldots \\
8 & \quad \frac{1}{6} & \quad \ldots \\
9 & \quad \frac{1}{6} & \quad \ldots \\
10 & \quad \frac{1}{6} & \quad \ldots \\
11 & \quad \frac{1}{6} & \quad \ldots \\
12 & \quad \frac{1}{6} & \quad \ldots \\
\end{align*}
\]

And = x \quad Or = +
Discrete Probability Distribution

- Can we separate the probabilities?
- $P(X_2 = \text{hot} | X_1 = \text{summer})$ is high; $P(X_2 = \text{hot} | X_1 = \text{winter})$ is low.
- $P(X_2 = v | X_1 = 1)$ means “Probability that roll 2 is $v$ if roll 1 is 1”
  - Independent variables

| $v$ | $P(X_2=v | X_1 = 1)$ |
|-----|----------------------|
| 1   | 1/6                  |
| 2   | 1/6                  |
| 3   | 1/6                  |
| 4   | 1/6                  |
| 5   | 1/6                  |
| 6   | 1/6                  |
Discrete Probability Distribution

- Using conditional probabilities allows us to easily combine:

\[ P(x, y) = \sum_{x} \sum_{y} P(x|y)P(y) \]

\[ P(y|x) = \frac{P(x|y)P(y)}{P(x)} \]

Bayes theorem
Putting it Together

• At time step 0:
  1. Robot takes an action
     \[ x^1 = f(x^0, u^0) + \eta^0 \]
  2. Robot makes an observation
     \[ z^1, h(x^1) \]

Distribution of measurement at time step 1:
Putting it Together

• At time step 0:
  1. Robot takes an action
     \[ x^1 = f(x^0, u^0) + \eta^0 \]
  2. Robot makes an observation
     \[ z^1, h(x^1) \]

Distribution of measurement at time step 1:
\[ p(x^1 \mid u^0, z^1) = \]
Putting it Together

• At time step 0:
  1. Robot takes an action
     \[ x^1 = f(x^0, u^0) + \eta^0 \]
  2. Robot makes an observation
     \[ z^1, h(x^1) \]

Distribution of position at time step 1:
\[ p(x^1 \mid u^0, z^1) \propto p(z^1 \mid x^1) \ p(x^1 \mid x^0, u^0) \ p(x^0) \]

Observation Model  Transition Model  Prior
Putting it Together

• At time step 1:
  1. Robot takes an action
     \[ x^2 = f(x^1, u^1) + \eta^1 \]
  2. Robot makes an observation
     \[ z^2, h(x^2) \]

Distribution of position at time step 1:
Putting it Together

• At time step 1:
  1. Robot takes an action
     \[ x^2 = f(x^1, u^1) + \eta^1 \]
  2. Robot makes an observation
     \[ z^2, h(x^2) \]

Distribution of position at time step 1:
\[ p(x^2 \mid u^0, z^1, u^1, z^2) = \]
Putting it Together

• At time step 1:
  1. Robot takes an action
     \[ x^2 = f(x^1, u^1) + \eta^1 \]
  2. Robot makes an observation
     \[ z^2, h(x^2) \]

Distribution of position at time step 1:
\[
p(x^2 \mid u^0, z^1, u^1, z^2) \\propto p(z^2 \mid x^2) p(x^2 \mid x^1, u^1) p(z^1 \mid x^1) p(x^1 \mid x^0, u^0) p(x^0)
\]

New trans. & obs. model Previous result
Putting it Together

• At time step 1:

  1. Robot takes an action
     \[ x^2 = f(x^1, u^1) + \eta^1 \]

  2. Robot makes an observation
     \[ z^2, h(x^2) \]

Distribution of position at time step 1:
\[
p(x^2 \mid u^0, z^1, u^1, z^2) \propto p(z^2 \mid x^2) p(x^2 \mid x^1, u^1) p(x^1 \mid u^0, z^1) \]

New trans. & obs. model Previous result
Recursive Inference

• At time step $t$:

1. Robot takes an action
   \[ x^t = f(x^t, u^t) + \eta^t \]

2. Robot makes an observation
   \[ z^t, h(x^t) \]

Distribution of position at time step 1:
\[
p(x^t | u^0, ..., u^t, z^1, ..., z^t) 
\propto p(z^t | x^t) p(x^t | x^{t-1}, u^{t-1}) p(x^{t-1} | u^0, ..., u^{t-1}, z^1, ..., z^{t-1})
\]

New trans. & obs. model
Previous result
Filtering Algorithm for Localization

- For each possible location:
  - Apply motion model
- For each possible location:
  - Apply observation model
- Loop forever
Filtering Algorithm for Localization

- For each possible location:
  - Apply motion model
- For each possible location:
  - Apply observation model
- Loop forever

- Too slow!
- Let’s use discrete hypotheses instead
Sampling From the Motion Model
Example Revisited

- Same flower-happy robot
- Same map
- This time, track samples (particles)
Example Revisited

Initially, no idea where we are

The particles. Height represents the weight (probability or confidence) of a given particle
Example

- First observation update
Example

- First observation update
- Evaluate model at particles
Example

New belief
Example

- New belief

[Diagram with flowers indicating 'too few' and 'too many']
Example

- Resample particles

Higher weight particles get more particles allocated near them during the resample
Example

- Reset weights

Density of particles is related to weight of particles previously
Example

- Robot moves, motion update

Particles spread out
Example

- Observation update
Example

Observation update
Example

- Estimate is best particle
Discrete Bayes Filter

**Algorithm 1: Discrete Bayes Filter Transition Update**

**Data:** Discretized state grid $X$

$N$-D probability map $m$

Action $u$

Copy $m_{Prev} \leftarrow m$;

for All states $x \in X$ do

\[ m[x] \leftarrow \sum_{y \in X} p(x|y, u) \cdot m_{Prev}[y]; \]

end

Normalize $m$;
Discrete Bayes Filter

Algorithm 2: Discrete Bayes Filter Measurement Update

Data:
Discretized state grid $X$
Landmark locations $L$
$N$-D probability map $m$
Observation $z$
Copy $m_{Prev} \leftarrow m$;
for All states $x \in X$ do
  $m[x] \leftarrow \sum_{l \in L} p(z|x, l) \cdot m_{Prev}[x]$;
end
Normalize $m$;