CS1951R: Intro to Robotics

Fly away with us!
Announcements

- PID project is out, due 11/13.
- Assignment 7: Transforms out Thursday; due 11/15
- UKF project was due 11/2.
- Tennis Court Day 12/4
- One more guest lecture!
  - John Kelly 12/6
- Hours tomorrow in CIT 115 from 10:30am-6pm EXCEPT
  - 3pm-4pm
  - 12pm-1pm
Course Structure

- Part I – Infrastructure.
- Part II – Sensing and State Estimation
- Part III – Control
- Part IV – Applications
Localization

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

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

We want to know: \( p(x_t \mid z_{1:t}, u_{1:t}) \)
Bayesian Filtering

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

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

State transition probability
Measurement probability


Algorithm 1 General Bayes Filter algorithm. For specific filters such as the Kalman Filter or the particle filter, the representation for bel and bel changes, and the corresponding mathematical updates take specific computational forms.

1: function PREDICT(bel(x_{t-1}), u_t, \Delta t)
2: \tilde{bel}(x_t) = \int p(x_t|u_t, x_{t-1})bel(x_{t-1})dx_{t-1}
3: return bel(x_t)
4: function UPDATE(\tilde{bel}(x_t), z_t)
5: bel(x_t) = \eta p(z_t|x_t)\tilde{bel}(x_t)
6: return bel(x_t)
7: function BAYESFILTER
8: u_t = COMPUTECONTROL(bel(x_{t-1}))
9: bel(x_t) = PREDICT(bel(x_{t-1}), u_t, \Delta t)
10: z_t = READSENSOR()
11: bel(x_t) = UPDATE(bel(x_t), z_t)
Monte-Carlo Localization

Algorithm MCL($\mathcal{X}_{t-1}, u_t, z_t, m$):
1. $\hat{X}_t = X_t = \emptyset$
2. for $m = 1$ to $M$
3. \hspace{1em} $x_t^{[m]} = \text{sample\_motion\_model}(u_t, x_{t-1}^{[m]})$
4. \hspace{1em} $w_t^{[m]} = \text{measurement\_model}(z_t, x_t^{[m]}, m)$
5. \hspace{1em} $\hat{X}_t = \hat{X}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$
6. endfor
7. for $m = 1$ to $M$
8. \hspace{1em} draw $i$ with probability $\propto w_t^{[i]}$
9. \hspace{1em} add $x_t^{[i]}$ to $\hat{X}_t$
10. endfor
11. return $X_t$

Table 8.2 MCL, or Monte Carlo Localization, a localization algorithm based on particle filters.
Monte-Carlo Localization
Monte-Carlo Localization

Figure 8.12 Illustration of Monte Carlo localization: Shown here is a robot operating in an office environment of size 54m × 18m. (a) After moving 5m, the robot is still highly uncertain about its position and the particles are spread throughout major parts of the free-space. (b) Even as the robot reaches the upper left corner of the map, its belief is still concentrated around four possible locations. (c) Finally, after moving approximately 55m, the ambiguity is resolved and the robot knows where it is. All computation is carried out in real-time on a low-end PC.
Localization

Where am I?

$x_t$: State at time $t$
$z_t$: Measurement at time $t$
$u_t$: Control input at time $t$

We want to know: $p(x_t|z_{1:t}, u_{1:t})$

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

State transition probability

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

Measurement probability
Localization

Where am I?

\( x_t : [x \ y \ \psi] \)
\( z_t : I_t \)
\( u_t : [\dot{x} \ \dot{y} \ \dot{z} \ \dot{\psi}] \)

We want to know: \( p(x_t|z_{1:t}, u_{1:t}) \)

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

State transition probability

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

Measurement probability
Measurement Model
Feature Extraction

- Input: an image.
- Output: keypoints in the same image.
- SIFT, SURF, BRIEF, FAST, HOG
- Intuition: look for gradients and edges.
- Some are rotation invariant; some aren’t.
Feature Extraction
Feature Matching

- **Input**: two images
- **Output**: feature correspondences between the two images
Estimate Rigid Transform

• Input:
  – Source matched points
  – Destination matched points

• Output:
  – Rigid transform between the source image and destination image.
Localization

Where am I?

\[ x_t: [x \ y \ \psi] \]
\[ z_t: I_t \]
\[ u_t: [\dot{x} \ \dot{y} \ \dot{z} \ \dot{\psi}] \]

We want to know: \( p(x_t|z_{1:t}, u_{1:t}) \)

\[ p(x_t|x_{0:t-1}, z_{0:t-1}, u_{1:t}) = p(x_t|x_{t-1}, u_t) \]
State transition probability

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

\[ p(z_t|x_t) \]
\[ p(I_t|x, y, z, \psi, m) \]
\[ p(f(I_t)|x, y, z, \psi, m) \]
\[ \prod_i p(r_t^i, \phi_t^i, s_t^i|x, y, z, \psi, m) \]

\[ h(i, x_t, m) = \begin{bmatrix} 
\sqrt{(m_{i,x} - x)^2 + (m_{i,y} - y)^2} \\
\text{atan2}(m_{i,y} - y, m_{i,x} - x) \\
s_i 
\end{bmatrix} \]
Localization

\[ p(z_t|x_t) \]

\[ h(x_t, m) = [x, y, z, \psi] \]
Localization in the Drone

Harris Corner Detector

- Uniform texture: pixel intensities don't change in all directions
- Positioned on edge: pixel intensities don't change along edge direction
- Positioned on corner: pixel intensities change in all directions
Localization

Where am I?

\[ x_t := [x \ y \ z \ \psi] \]
\[ z_t := I_t \]
\[ u_t := [\dot{x} \ \dot{y} \ \dot{z} \ \dot{\psi}] \]

We want to know: \[ p(x_t | z_{1:t}, u_{1:t}) \]

\[ p(x_t | x_{0:t-1}, z_{0:t-1}, u_{1:t}) = p(x_t | x_{t-1}, u_t) \] State transition probability

\[ p(z_t | x_{0:t}, z_{1:t-1}, u_{1:t}) = p(z_t | x_t) \] Measurement probability
Dynamic Environments

Figure 8.20 Comparison of (a) standard MCL and (b) MCL with the removal of sensor measurements likely caused by unexpected obstacles. Both diagrams show the robot path and the end-points of the scans used for localization.
Localization
Natural Language Command of an Autonomous Micro-Air Vehicle
Albert Huang, Stefanie Tellex, Abe Bachrach, Thomas Kollar, Nick Roy
Oculus Go
Microsoft HoloLens
The Magic Leap One mixed reality headset is shipping today for $2,295

By Adi Robertson | @thedextriarchy | Aug 8, 2018, 8:08am EDT