Localization, Mapping and Exploration with Multiple Robots

Dr. Daisy Tang
Two Presentations

  - Presented by Brent Wasilow

  - Presented by Kushal Patel
Mapping

Mapping

- Concurrent mapping and localization problem
  - Building maps when a robot’s locations are known is relatively straightforward
  - Localizing a robot when a map is available is also well-understood
  - The combination makes it hard

Challenge

- When mapping environments with cycles, the robot’s cumulative error can grow without bounds
Basic Idea

- Combine the idea of posterior estimation with incremental map construction using maximum likelihood estimators.

- An algorithm that builds large maps in real-time on a low-end computer.

- Posterior estimation can integrate data collected by more than one robot.

- Can be extended to 3D mapping.
Likelihood Function

- Concurrent mapping and localization = maximum likelihood estimation problem
- Let m be a map, which is a collection of scans and their poses (x-y-θ)

\[ m_t = \{ (o_\tau, \hat{s}_\tau) \}_{\tau=0,\ldots,t} \]  \hfill (1)

- The goal is to find the most likely map given the data

\[ \arg \max_m P(m \mid d_t) \]  \hfill (2)

\[ d_t = \{ s_0, a_0, s_1, a_1, \ldots, s_t \} \]  \hfill (3)

The set of values for m for which \( P(m \mid d_t) \) is maximized
\[ P(m \mid d_t) = \eta P(m) \int \cdots \int \prod_{\tau=0}^{t} P(o_\tau \mid m, s_\tau) \cdot \prod_{\tau=0}^{t-1} P(s_{\tau+1} \mid a_\tau, s_\tau) \, ds_1 \ldots ds_t \]
Odometry/Encoders

- **Purpose**
  - To measure turning distance of motors (in terms of rotations), which can be converted to robot translation/rotation distance
  - If wheel size known, number of motor turns → number of wheel turns → estimation of distance robot has traveled

- **Basic idea in hardware implementation:**

  ![Diagram](image)

  Device to count number of “spokes” passing by
Encoders

**Challenges/issues:**

- Motion of wheels not corresponding to robot motion, e.g., due to wheel spinning
- Wheels don’t move but robot does, e.g., due to robot sliding
- Error accumulates quickly, especially due to turning:

  ![Odometry Data](image)

  - **Red line** indicates estimated robot position due to encoders/odometry/dead reckoning.
  - Begins accurately, but errors accumulate quickly.
Motion Model

- This figure depicts the probability of being at pose $s$, if the robot previously was at $s'$ and executed action $a$
- The distribution is obtained by kinematic equations, assuming robot motion is noisy along its rotational and translational component
Perceptual Model

- Inherited from scan matching and projection filtering
- When a robot receives a sensor scan, it is unlikely that future measurements perceive an obstacle within space previously perceived as free

Figure 3: Likelihood function generated from a single sensor scan. The robot is on the left (circle), and the scan is depicted by 180 dots in front of the robot. The likelihood function is shown by the grey shading: the darker a region, the smaller the likelihood for sensing an object there. Notice that occluded regions are white (and hence incur no penalty).
Motion model and perceptual model are differentiable.

Their approach uses the gradients for efficiently searching the most likely pose of a robot given its sensor measurements.

1000 or more gradients per second can be computed on a low-end PC.
Conventional Incremental Mapping

- Given a scan and an odometry reading, determine the most likely pose. Then append the pose and scan to the map, and freeze it once and forever.
- Typically works well in non-cyclic environments.

\[
\hat{s}_t = \arg\max_{s_t} P(s_t | o_t, a_{t-1}, \hat{s}_{t-1}) \quad (6)
\]

\[
m_{t+1} = m_t \cup \{\langle o_t, \hat{s}_t \rangle\} \quad (7)
\]

- When closing cycle, however, this approach suffers from two shortcomings:
  - Pose errors can grow arbitrarily large.
  - When closing a loop in a cyclic environment, past poses may have to be revised to generate a consistent map.
Incremental Mapping Using Posteriors

- This approach computes the full posterior over robot poses, instead of the maximum likelihood pose only.
- It is a probability distribution over poses
  \[ Bel(s_t) = P(s_t \mid d_t, m_{t-1}) \]  \hspace{1cm} (8)
- The initial belief \( Bel(s_0) \) is centered on an origin.
- The new belief is obtained through:
  \[ Bel(s_t) = \eta P(o_t \mid s_t, m_{t-1}) \int P(s_t \mid a_{t-1}, s_{t-1}) \, Bel(s_{t-1}) \, ds_{t-1} \]  \hspace{1cm} (9)
- After computing the belief, the new map is generated by:
  \[ \bar{s}_t = \arg\max_{s_t} Bel(s_t) \]  \hspace{1cm} (11)
  \[ m_{t+1} = m_t \cup \{o_t, \bar{s}_t\} \]  \hspace{1cm} (12)
Sample-Based Approximation for Belief

Figure 5: Sample-based approximation for the posterior $Bel(s)$. Here each density is represented by a set of samples, weighted by numerical importance factors. Particle filters are used to generate the sample sets.
Example

$M$ population of particles

Each particle has an importance weight

Increased # of particles around locations
This approach performs gradient descent using each sample as a starting point, then computes the goodness of the result using likelihood function.

If samples are spaced reasonably densely, one can guarantee the global maximum of the likelihood function.

Different from a single starting pose, which might fail to produce the global maximum.
Backwards Correction

- When closing cycles it is imperative that maps are adjusted backwards in time, given by the difference
  \[ \Delta s_t = \bar{s}_t - \hat{s}_t \]  
  \[ (13) \]
- **Difference** between the incremental best guess and the best guess using full posterior

- First, *size of the loop* is determined by the scan in map
- Second, the *error is distributed* proportionally among all poses in the loop
- Finally, *gradient descent search* is applied iteratively for all poses, until the map is maximally consistent
Multi-Robot Mapping

- Assume initial pose of robots relative to each other is unknown.
- Each robot starts within the map of a specific robot (team leader).
- To generate a single, unified map, each robot must localize itself in the map of the team leader.
Experiments: Robots
Mapping A Cycle

- Scans were appended to the map when robot moved 2 meters
- Manageable complexity of maps
- When cycle is closed (6b), error is significant; true pose is quickly identified and reduces uncertainty (6c)
Mapping Without Odometry

- Build the same map in the absence of odometry data
- Final result is optically equivalent to the one with odometry
- Odometry-free results are only possible if the environment possesses sufficient variation
Autonomous Exploration

- Urban robot, skid-steering mechanism using tracks, extremely erroneous

**Figure 9:** Autonomous exploration and mapping using the urban robot: Raw data and final map, generated in real-time during exploration.
Conclusion

- This approach combines ideas from incremental mapping (maximum likelihood, incremental map construction) with ideas of non-incremental approaches (posterior estimation, backwards correction)
- The result is a fast and robust algorithm for real-time mapping of indoor environments, which extends to multi-robot mapping