Computational Principles of Mobile Robotics

Mapping and related tasks

Mapping and related tasks

- Extremes
  - Metric – embedded in some space with an underlying metric (typically a Cartesian map).
  - Topological – no metric information (Graph-like).
  - Intermediate forms
    - Embedded graphs, etc.

Babylonian Map of the World. Circa 9th century BC. Oldest known map of the world

Layers of map

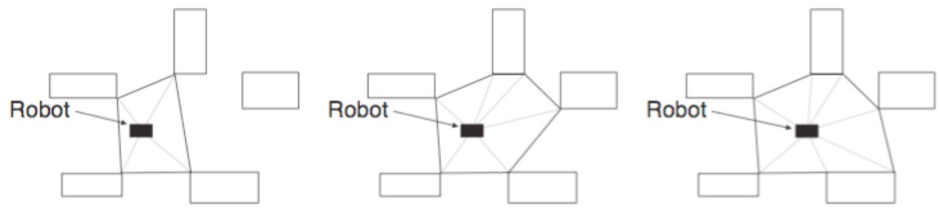
- Sensoral – raw data signals.
- Geometric – 2D or 3D objects.
- Local relational – functional, structural.
- Topological – large-scale relational
- Semantic – functional labelling.

10.1 Sensorial maps

- Maps based directly on sensor measurements
  - $\{I_i(x_i, y_i, \theta_i)\}$
  - Use this to estimate $I(x, y, \theta)$
- Has the advantage of being very general, but in practice it may be difficult to build this representation directly.

10.2 Geometric maps

- Construction of a geometric primitive-based map.
- Edge/surface representations are common.
- Representation can be general or more restrictive.
  - Street or generalized-street



(a) Step I (b) Step II (c) Step III

10.2.1 Mapping without localization

- Basic idea (Elfes and Moravec) is to construct an occupancy grid representing the probability a location z is occupied given sensor measurements.
- A Bayesian updating mechanism is used to update probabilities based on an appropriate sensor model and priors for occupancy and sensor.

$$p(z|r) = \frac{p(z)p(r|z)}{p(r)}$$

Suppose we get a measurement r, what is the probability of there being an object at z?

Mapping without localization

- So every measurement can be used to update the probability of an object at a given point
  - Need priors for
    - p(r) probability of a reading and
    - p(z) probability of some place holding an obstacle
  - And the conditional probability p(r|z)
    - Which is basically a model of sensor performance
  - Choose a MAP (maximum a posteriori) estimate

Of course, mapping without localization does not occur frequently

10.2.2 Simultaneous localization and mapping

- Goal – construct a conditional pdf that describes the trajectory of the robot (s_t) and a static map Θ given control inputs u^t, robot measurements z^t and data associations n^t

$$p(s_t, \Theta | z^t, u^t, n^t).$$

- If we make reasonable assumptions about independence over time and that various processes are Markovian, then

$$p(s_t, \Theta | z^t, u^t, n^t) = \eta p(z_t | s_t, \Theta, n^t) \int p(s_t | s_{t-1}, u_t) P(s_{t-1}, \Theta | z^{t-1}, u^t, n^t) ds_{t-1}$$

Simultaneous localization and mapping

- This Bayesian SLAM filter can be solved in many ways.
- Two basic approaches
  - Kalman filter – strong assumptions about error distributions of plant and sensor errors.
  - Particle filter – issues related to ensuring appropriate sampling of the distribution functions.

SLAM: Kalman filter approach

- Assume n map features (n increases over time) and a point robot

$$\boldsymbol{x}(k) = [\ x \quad y \quad p_x^1 \quad p_y^1 \quad p_x^2 \quad p_y^2 \quad \ldots \quad p_x^n \quad p_y^n\ ]^T$$

- Plant

$$\begin{aligned}
\boldsymbol{x}(k+1) &= \boldsymbol{\Phi x}(k) + \boldsymbol{\Gamma u}(k) + \boldsymbol{v}(k) \\
\begin{bmatrix} x(k+1) \\ y(k+1) \\ p_x^1(k+1) \\ p_y^1(k+1) \\ p_x^2(k+1) \\ p_y^2(k+1) \\ \ldots \\ p_x^n(k+1) \\ p_y^n(k+1) \end{bmatrix} &= \begin{bmatrix} x(k) + \triangle x(k) + v_x(k) \\ y(k) + \triangle y(k) + v_y(k) \\ p_x^1(k) \\ p_y^1(k) \\ p_x^2(k) \\ p_y^2(k) \\ \ldots \\ p_x^n(k) \\ p_y^n(k) \end{bmatrix}
\end{aligned}$$

SLAM: Kalman filter approach

- Assume a sensor model that provides some estimate of the robot's motion and of the displacement to the environmental features (the map).
- Kalman filter can 'solve' for robot state and marker position.
  - Need to do data association.
  - Size of the problem grows with the addition of (new) features.

SLAM: Particle filter approach

- A *pure* particle filter would use a single particle filter to estimate the SLAM posterior.
  - Impractical due to the dimensionality of the problem.
- More commonly a Rao-Blackwellization process is sued to reduce the dimensionality of the problem
  - FastSLAM and DP-SLAM as examples
- But let us consider the pure problem here

SLAM: Particle filtering approach.

- The SLAM posterior at time k is represented by a set of sample S_k^i that is $p(s_t, \Theta | z^t, u^t, n^t)$
  - Resampling phase. Resample the distribution so that the particles well represent the distribution.
  - Prediction phase. Each particle moves forward in time simulating $p(s_t | s_{t-1}, u_t)$
  - Updating phase. Sensor measurements are used to update the weights associated with each particle $p(z_t | s_t, \Theta, n^t)$
- Use these particles to compute

$$p(s_t, \Theta | z^t, u^t, n^t) = \eta p(z_t | s_t, \Theta, n^t) \int p(s_t | s_{t-1}, u_t) p(s_{t-1}, \Theta | z^{t-1}, u^t, n^t) ds_{t-1}$$

SLAM: Particle filters

- This pure version is not always used in practice. Rather, effective approaches use a combination of Rao-Blackwellization to reduce the dimensionality of the probability space and/or use Kalman filters to model the map features.
- DP-SLAM and DP-SLAM 2.0 are pure particle filtering approaches.

10.2.3 Loop closing

- Error grows over time, so a critical question in any SLAM algorithm is ‘have I been here before’.
  - Mathematically this is embedded in the data association term (does this measurement belong to that map feature).
- Thus loop closing is a critical problem
  - If we can (with certainty) say that a loop has/has not closed then part of the probability function will collapse and does not need to be represented.
- Particle filters can retain both solutions.
- Kalman filters must make the decision immediately.

10.2.4 Factor Graphs and GTSAM

- As the SLAM process becomes more complex the process of properly representing the relationships between the various factors becomes more difficult.
- One effective approach to dealing with this complexity is through the use of factor graphs to represent these relationships.
- GTSAM provides a library to support this.

10.3 Topological maps

- SLAM can be expressed within a topological framework
  - Either separately or as part of some hybrid representation.
- A pure topological representation allows basic properties of SLAM ot be examine without dealing with details of data association, etc.
- Under reasonable assumptions it is not possible to solve SLAM deterministically without resorting to aids of some kind.

10.3.1 Marker-based exploration

- If a robot is equipped with a single unique moveable marker it can solve the SLAM problem deterministically
  - Cost is quite high in terms of motion complexity O(mn).
- A single fixed unique marker is not sufficiently powerful, although a single fixed unique directional marker **is.**

10.4 Exploration

- In any SLAM algorithm the robot must decide where to explore next.
- There may exist large ‘unknown’ areas, choice of which to explore can have significant impact on overall algorithm performance.

10.4.1 Spiral search

- Simplest version involves searching a line
  - Try in one direction a distance d
  - If foud, done. Otherwise move back a distance 2d
- Repeat until found
- Can show that if the target is a distance d form the start, you will find it within 9d