Mobile robotics II

Key points:
- Topological and geometric maps
- Varieties of spatial representations
- Relating sensing to space
- Localisation: determining and updating
- Learning maps
- Simultaneous localisation and mapping

Topological maps
- Can represent landmarks and the routes between them as nodes and edges in a graph
- Edges can represent motor action needed to get from one node to the next, or direction, distance, path convenience etc.
- Can then use standard AI graph search methods to find route from start to goal

Geometric maps
- Can describe location of robot and objects in its world as a configuration space
- For mobile robots, usually collapse 6 d.o.f. to 2 d.o.f.
  - Assumes robot moves on ground plane
  - Can rotate on spot (so direction not important)
  - Only obstacle location matters
- Expand actual obstacles by robot size
- Navigation then involves finding routes through freespace

Grid representation
- Can use ‘wave front’ to find route

Voronoi diagram
- Convert free space to graph by e.g. skeletonisation – edges are equidistant locations from obstacles
- Can find routes using graph search, as for topological map

Figures from Murphy (2000)

Advantages of topological maps:
- Only sparse data storage
- Representation matches problem description: e.g. instruct robot to move between discrete locations
- Recognition only requires consistency, not accuracy

Advantages of metric maps:
- Can extrapolate between known locations
- Can derive novel shortcuts
- Common representation to fuse sensor/motor data

[Diagram showing advantages of topological and metric maps]

Relating sensing to space
- Some sensors provide metric information almost directly e.g. range finders
- Others are good for distinctive landmarks but hard to convert to metric layout e.g. vision
- Main problems are
  - Aliasing (different places sometimes look the same)
  - Variability (same place sometimes looks different)

Relating sensing to space
- Tracking movement (commands or odometry) to know where you are can help against aliasing and variability problems
- Recognising where you are from external cues can correct for the cumulative error of dead-reckoning
- Note that metric approach fuses the two sources of data in common representation

Localisation
- ‘Lost robot problem’: can you recognise where you are when switched on?
- Alternatively, given model of environment, what position or node is the most probable, based on the current sensory input?
- Essentially same as visual recognition problem in previous lectures, and can use same Bayesian approach i.e.

\[
p(s|z) = \frac{p(z|s)p(s)}{\sum p(z|s)p(s)}
\]

where s is robot location and z is sensor input and have been given or have learned \(p(z|s)\) and \(p(s)\)
Updating the localisation estimate

- ‘Lost robot’ could also use active means to confirm location – i.e. does movement produce the expected sensory consequence?
- More generally, current position estimate can be function of previous estimate, expected result from moving, and current input
- Suggests a Kalman filter approach: estimate current state of system based on previous state, model and measurement, all of which are noisy

Simultaneous localisation and mapping

- Usually not true that robot knows (with certainty) its position while moving around to build the map
- But map built so far can be used to help correct localisation estimate

Aim to merge all past sensor data: \( z' = z_0, z_1, ..., z_t \)

and movement data: \( u' = u_0, u_1, ..., u_t \)

\( p(s_t, m | z', u') = \frac{p(s_{t-1}, m | z', u')}{\text{normalisation factor}} \)

Learning the map

E.g. Occupancy grid approach:
Assuming robot knows where it is in grid, sensory input provides noisy information about obstacles, e.g. for sonar

\[
\text{probability of given sonar measurement } (z) = \frac{1}{R^r} \left( \frac{\beta - \alpha}{\beta} \right)^{n} \text{ for } n \text{ grid elements in region 1 is occupied } (O)
\]

Using Bayesian approach

\[
p(O | z) = \frac{p(z | O) p(O)}{p(z | O) p(O) + p(z | -O) p(-O)}
\]

where \( p(O) \) will depend on previous measurements

Variants on SLAM:

Expectation maximisation (e.g. Thrun et al 1998)
- Deals explicitly with aliasing: uses hill-climbing to find the most likely map based on the possible paths of the robot given the sensor data
- Makes no assumptions about noise distributions

But
- Only weak convergence result
- Runs off-line, i.e. gather all data then process
- Only maintains single maximum likelihood map

Further reading

Summary

- There are sound theoretical approaches to path planning, localisation, map building and SLAM
- Problems lie in application to real, limited, noisy, dynamic robots and environments
- Best approach is then dependent on task, sensors and actuators
- ‘Hybrid’ solutions are often useful