Past, Present, and Future of Simultaneous Localization and Mapping: Towards the Robust-Perception Age (as summarized by Kevin Tan)

Introduction

SLAM is the simultaneous estimate of a representation of the environment (the map) and the position of the robot in that environment. It creates maps that are useful for both downstream tasks like path planning but also for limiting the localization error (improving it from dead-reckoning by, say, integrating wheel encoders). Its popularity is connected to the rise of mobile indoor robotics where GPS is not available to bound the localization error. We implemented EKF and Particle Filter SLAM for homework.

Anatomy of a Modern SLAM System

There are two main parts to a modern SLAM system: the front end and the back end.

1. The **front end** is in charge of taking in raw sensor measurements, extracting features from them, and associating those features to landmarks in the environment (data association).
2. The **back end** is in charge of taking the abstract, higher-level features from the front end and using it to perform inference on a factor graph representation of all the different variables in the environment that the robot needs to estimate (which includes the robot's pose at different points in time, landmark locations, and camera intrinsic parameters).

Robustness of SLAM Systems

Some of the main failure modes for modern SLAM systems include incorrect data association[1] where measurement-landmark matches are wrong (a task that is generally given to the SLAM front end), a highly dynamic environment (because SLAM generally makes the *static world assumption*), and hardware aging where the sensor parameters during initial map construction are no longer similar to the sensor parameters during localization.

Making SLAM Systems Scalable

As robots have increasingly large operation domains and long operation times, the factor graphs that they build and perform inference on grow unboundedly large. In order to make SLAM systems that work in these situations, sparsification, out-of-core, and multi-robot methods have been used. Sparsification leverages the insight that not all landmarks are created equal, and devises strategies to only add information-rich landmarks. Out-of-core methods distribute factor-graph solving to multiple processors, and multi-robot technique divides the mapping of a large area across more than one robot.

Representing Maps

Deciding how to representing the environment is standardized in the 2D case with landmark-based maps (like in our assignments) and occupancy grid maps (like in our final project). In the 3D case, there is much more disagreement. There are landmark-based sparse representations, low-level raw dense representations, boundary and spatial-partitioning dense representation, and high-level object-based representations. Personally, I find the last one the most exciting in that future SLAM systems could potentially build highly expressive and interpretable maps that can power extremely complex and high-level decision making.

Deep Learning

This tool is rapidly changing classical robotics. In particular, it shows promise in connecting raw sensor data to either understanding of the environment or direct control actions. It may be possible one day to create an end-to-end SLAM system using a deep neural network without explicit feature modeling and data association. Also, in our final project, we are using a CNN (type of neural net) for object detection!

---

[1] Data association can be further broken up into the *short term* and *long term* data association. An example of short term data association would be "given temporally close sensor measurements, which readings correspond to the same landmark?" An example of long term data association would be "given temporally distant sensor measurements, which readings correspond to the same landmark?"