Environmental Text Assisted Mixture Monte Carlo Localization

Proposed Problem

- Most humans use visual cues – pictographs, text, and distinct local objects/features – to localize within a known environment, whereas most of the more robust implementations of robot localization rely instead on precise wide-angle planar laser scans of the environment.
- The goal of this project is to attempt to develop a more human-oriented MCL system capable of using text extracted from the environment instead of wide-angle planar laser scans.

Solution Design

- Mixture Monte Carlo Localization
  - Forward Monte Carlo Localization
  - Dual of Monte Carlo Localization
  - Environmental Text Modifications
    - Text Tagged Mapping System
    - Environmental Text Extraction and Character Recognition

Forward Monte Carlo Localization

- Generate a new particle by sampling over the cumulative distribution of the last particle set's weights.
- Apply odometry transformation to the particle to determine new particle pose within the map.
- Calculate weight of the particle as a function of the distance sensor scan and the rays traced from the new pose to the map walls.

Dual of Monte Carlo Localization

- Generate a new particle by randomly creating a particle consistent with the distance sensor scan (the more information with each distance scan, the better the particle generated).
- Apply odometry transformation to particle to determine its pose if it had existed in the last time step (set of particles).
- Calculate weight of the particle as a function of the distance sensor scan and the rays traced from the 'previous' pose to the map walls.

Mixture MCL

- With some small probability, phi, the system will generate a sample for the new particle set using DMCL, otherwise the sample is generated using FMCL. The process repeats until the particle population limit is reached or the sum of all particle weights exceeds a threshold.
- There are better methods for adapting population size to the uncertainty of localization, but time constraints intervened.

Environmental Text Augmentations

- Can supplement both forward and dual of Monte Carlo Localization.
- FMCL: Including identified text tags in the particle weight calculations.
- DMCL: Using a text tagged map to identify regions to concentrate particle generation.
- If there is no text detected in the scene in the last time step, then FMCL and DMCL use their normal weight functions.

Environmental Text Processing

- Text Extraction
  - Stroke Width Transform
    - “Detecting Text in Natural Scenes with Stroke Width Transform.” Epshtein et al., 2010.
    - “read_text” ROS node
- Character Recognition
  - tesseract-ocr
- Text Tag Matching
  - Dictionary search of the map's taglists

Environmental Text Extraction

- Stroke Width Transform as implemented by Menglong Zhu as the “read_text” ROS node - the main component of the literate_pr2 stack.
- SWT Algorithm
  - Canny Edge Detection to find text candidates.
  - Determine direction of gradient at edges.
  - Ray tracing at all edge pixels to determine “stroke width” of pixels within object.
  - Elimination of objects with excessive variance in stroke width (non-text).

Environmental Text Recognition

- After SWT locates likely text in a scene, the objects are extracted from the scene and sent to Tesseract-OCR (Optical Character Recognition).
- If the object's region of interest contains discernible text at sufficiently high resolution, then tesseract will return the text as a string.

Additional Environmental Text Processing

- All extracted text from a single image is combined into a data structure containing each string, the size and location of each region (in pixels), the approximate distance of each region from the camera (in meters measured by either a Sharp IR sensor or a Kinect depthmap), and a time stamp of the sensor readings.
- This structure is processed to estimate the height of each text object and its location relative to the camera.

Map Tagging Architecture

- Each Map object contains a map array, the map's resolution, and a vector of TagLists.
- Each TagList contains a weight and a vector of Tags.
- Each Tag contains a UUID, a text string, the font height, the coordinates of the text on the map, and a timestamp of creation.
  - The UUID and time stamp permit multiple locations for a single text string (exit signs, first aid kits, etc.).

Text Tag Weighting and Seeding

- The processed text structure is passed to the map's dictionary search, which iterates through all Tags within the Map's TagLists and calculates the weight of each possible match as a function of the TagList weight, the height of the text in the Tag, the height of the extracted text, and the number of character matches within the two text strings.
- These Tag matches are then used with the text distance and orientation relative to the camera within the FMCL weight function and for DMCL particle generation.

Demonstration

- It was intended to have a working series of ROS nodes and a video demonstration of the system in action, but the complexity of the design and degradation of my programming ability were both underestimated.
- Also, the omni-wheeled rover has no ROS support, very poor odometry, and non-rounded rollers.

Future Work

- Very hopeful that there will be a working system to test within a few days.
- If a working system is finished, it will be uploaded to github.
  - github.com/tician