1 Dates

Assigned: September 13, 2005 NEW DUE DATE: September 25, 2005, 11:59 PM

2 Introduction

In the previous lab, you became acquinted with running and writing clients for Player/Stage/Gazebo (PSG). In the upcoming weeks, you will write robot controllers for:

1. Localization: inferring a robot's location on a map from its sensory information
2. Path planning: moving the robot to specified locations on map
3. Exploration/Coverage: controlling a robot to explore and cover an unknown environment

3 Assignment

For this project, you are expected to create new Stage and Gazebo worlds and develop client programs to drive and localize Pioneer robots. The core component of the project, robot localization, will require the implementation of a particle filter.

The first step for this project is to create maps (as PPM images) and terrains (using the gzbuilder utility) to incorporate into robot world files. This would should include a Pioneer2DX robot. You are welcome to use example world and config files from the lab and /contrib/projects/psg/ directory. From the previous lab, you should already be familiar with routines for incorporating map and terrain data with PSG.

In the particle framework, your are estimating the state of the robot, expressed in 3D as an X,Y location and a angular heading, from observed laser range and odometry readings. For details about particle filter implementation, we refer you to the Condensation paper [1] (particularly Figures 5 and 6). You should pay particular attention to your likelihood function, that is, how the probability of observations given a robots location. Towards your likelihood function, we provide you a PPM image reader and writer, available from the /courses/cs148/asgn/atrack/mcl as Image.C and Image.H. Map files as PPM images can be read and edited using these routines. To use these image routines, you must write your client in C++ using the Player C++ client library, documentation available at:

http://playerstage.sourceforge.net/doc/Player-cvs/player/client_libs.php

You are welcomed to use any other client library, but you will be responsible for finding your own PPM support and writing an accessible Makefile.

3.1 Implementation

During this assignment you will implement:

- a simple exploration or teleoperation client for Player
- a localization routine for your client
  - your computation must use only robot sensor data. More specifically, global and absolute truth information, such as position→px, **cannot be used for your computations**. However, absolute truth information can be used for recording the performance of the robot and evaluating offline.
  - the robot should be using some form of noisy odometry
  - the robot client must record estimated and actual locations visited in the environment in a form that can be displayed after program execution.
  - the robot client must provide a command line parameter to specify time allowed for execution in milliseconds. When this parameter is specified, the robot client must terminate after the specified time has elapsed.
- an interesting environment to test the localization capabilities of your client

4 Deliverables

Once completing this implementation, you will turnin the following:

1. the **commented** source code and compiled executable for your client program
   - usage for the client must be obtainable from the command line and the source file header
   - build instructions should be available from the source file header
   - one source file is preferred, but not required
2. directions, scripts, or source for displaying coverage after client execution
3. world files, images, and other necessary files for your Stage and Gazebo worlds
4. a project writeup (refer to course missive)
5. refer to the course missive for the format of the electronic submission
   - one suggestion: a plot showing area covered by a robot over varying execution durations and initial starting locations

4.1 Electronic Submission

You will submit your project deliverables via the cs148 handin script. The identifier for this project will be "advanced_mcl". The handis script can be invoked by the following:

```
>> cs148\_handin advanced\_mcl <directory to handin>
```

Refer to the missive for proper structuring of your project submission.

5 References

1 M. Isard, A. Blake, "CONDENSATION – conditional density propagation for visual tracking", Int. J. Computer Vision, 29, 1, 5–28, 1998, http://www.robots.ox.ac.uk/ ab/abstracts/ijcv98.html

2 PSG User Documentation, http://playerstage.sourceforge.net/doc/doc.html

3 Libplayerc Client Library Reference, http://playerstage.sourceforge.net/doc/Player-cvs-html/libplayerc/

4 Player C++ Client Library, playerstage.sourceforge.net/doc/Player-cppclient-1.5.pdf Libplayerc