Massachusetts Institute of Technology

Robotics: Science and Systems – Spring 2005

Visual Navigation with Fiducials

Thursday, April 14

Objectives

This supplemental handout describes the use of known fiducial objects to infer robot pose. You should also refer to your handouts and notes from Lectures 9 (Localization) and 10 (Mapping).

1 Map and Landmark Format

We describe a visual navigation capability that depends upon a map of the environment in which the robot is to operate. We will provide a physical maze, and a "maze map" describing the structure of the maze. The maze map is an ascii file with one-character keywords, integer IDs for vertices, edges, and balls, real-valued locations (in meters), real-valued landmark attributes (color in HSB, diameter in meters), and the comment character #.

Maze files `maze.dat` will have the following format for this Lab and for the remainder of the term. In the example below, "V," "E" and "B" are one-character keywords, "N," "M," "K" and all "ids" are small integers, and all other values (coordinates, hues, saturations, diameters) are real-valued. The character # is a comment character whose meaning is "ignore until end of line."

```
# the N vertices, with integer IDs and xy coords for each
# origin is arbitrary; x and y should be in meters
# vertex id's should be non-negative and distinct, but need not be monotonic
V vid1 x1 y1
V vid2 x2 y2
...
V vidN xN yN

# the M edges, with integer IDs and vertex IDs for each
# edges are ordered so that they contain robot area to their left
# i.e., the edge from 0,0 to 0,1 contains the halfplane x <= 0
# edge id's should be non-negative and distinct, but need not be monotonic
E eid1 vid_i1 vid_j1
E eid2 vid_i2 vid_j2
...
E eidM vid_iM vid_jM

# the B balls,
# with integer IDs, 3D position, color, and radius in meters
# ball id's should be non-negative and distinct, but need not be monotonic
B bid1 ball_x1 ball_y1 ball_z1 ball_Hue1 ball_Sat1 ball_rad1
B bid1 ball_x2 ball_y2 ball_z2 ball_Hue2 ball_Sat2 ball_rad2
...
B bidK ball_xK ball_yK ball_zK ball_HueK ball_SatK ball_radK
```

2 Visual Navigation

This section guides you through the modification and extension of your VisualServo Carmen module (or equivalent) from Lab 4, to enable your robot to infer its approximate position and heading (i.e., absolute orientation) in maze coordinates. As visual landmarks, your robot will rely on small colored balls mounted on dowels above the maze walls. Each landmark's diameter, color (hue and saturation), maze location (x, y), and height (z) are included in the maze map described above.

Recall that you implemented a calibrated range sensor and uncalibrated bearing sensor using a calibrated camera in Lab 4. For visual navigation, to extend your sensor so that it returns the range and calibrated, body-relative bearing of each observed landmark. Using two such landmark observations, your robot can determine its position and (absolute) heading in maze map coordinates. When three or more landmark observations are available, your robot can use the additional constraints to combat noise, for cross-validation, or for both purposes.

2.1 Bearing Camera Calibration

Position the camera so that it is centered left-to-right on the robot. Orient it so that it is looking straight ahead. (Think about how you can use external markings, e.g., ruled lines on the floor, to help you do this.)

Observe a gradated meter stick or measuring tape. Make sure that the tape is aligned with the middle horizontal raster of your image. Make sure that the 1m mark is at the center pixel, and that the number of pixels subtended by the left segment is the same as that by the right segment. This ensures that your calibration object lies in a plane very nearly parallel to the image plane (do you see why?).

Now measure the perpendicular standoff distance from the camera to your calibration object, and work out the relationship between pixel coordinate x and body-relative bearing angle α.

2.2 Solving for Robot Pose

Combining your existing blob tracker, and the additional bearing calibration information from the previous section, yields a metric range-and-bearing sensor for any detected fiducial, provided you know the actual dimensions of the fiducial in the scene. Refer to your course notes to compute the possible location(s) of the robot given two range-and-bearing observations.

Remember to take into account the height (z coordinate) of the fiducial, and the height of your camera's focal point above the ground plane ($z = 0$ in map coordinates).