Robot Assignment

951926 1001231 1024072 1028907

***Abstract*—In this report, we present our solution to the 1996 AAAI Mobile Robotics Competition task "Call a meeting"[1]. We provide some background information about the areas of robotics that are relevant to the problem, with brief reference to literature. We then provide a detailed description of our system and evaluate its performance. Finally, we discuss the experimental results and suggest areas for improvement.**

I. Background

A. Robot Motion

holonomic robots, PID control, reactive control

B. Localisation

- odometry issues
- beliefs, prior and posterior $\overline{bel}$
- sensor model, update
- motion model

The aim of localisation is to obtain an estimate of the position of the robot at any given time. While it is possible to use odometry data to do so, this data is inherently noisy and prone to error. In particular, odometry errors can arise from changes in the surface being traversed and the robot's weight, among others. It is possible to mitigate the effect of these errors on estimates of the robot's position by using Bayesian techniques. Bayes filter, Kalman filter, particle filter (MCL), small section about mapping—still an active area of research in robotics. Mention SLAM, which has been pretty much solved.

Algorithm 1 Basic Monte Carlo Localisation[2]

Algorithm MCL$(\mathcal{X}_{t-1}, u_t, z_t, m)$
$\bar{\mathcal{X}}_t = \mathcal{X}_t = \emptyset$
for $m = 1$ to M **do**
  $x_t^{[m]} =$ **sample_motion_model**$(u_t, x_{t-1}^{[m]})$
  $w_t^{[m]} =$ **sensor_model**$(z_t, x_t^{[m]}, m)$
  $\bar{\mathcal{X}}_t = \bar{\mathcal{X}}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$
end for
for $m = 1$ to M **do**
  draw i with probability $\propto w_t^{[m]}$
  add $x_t^{[i]}$ to $\bar{\mathcal{X}}_t$
end for
return $\bar{\mathcal{X}}_t$

C. Route Planning

PRM (sampling methods, graph search), RRT

D. Exploration

frontier based techniques

Algorithm 2 Probabilistic Road Map Generation

Algorithm generate_PRM(map)
$V =$ **sample_vertices**(map)
for $v_c \in V$ **do**
  while $c(v_c) < C$ **do**
    $v_t =$ **get_closest**(V)
    if $d(v_c, v_t) < D_n$ **then**
      if $\neg\phi(v_c, v_t) \wedge \gamma(v_c, v_t)$ **then**
        connect(v_c, v_t)
      end if
    else
    end if
  end while
end for

Algorithm 3 Path Flattening

Algorithm flatten_path(P, I, map)
for $i := 0$ **to** I **do**
  for $j := 0$ **to** $|P| - 2$ **do**
    $A \leftarrow$newpath(i)
    $B \leftarrow$newpath$(i+1)$
    $C \leftarrow$newpath$(i+2)$
    if freely_connected(map,A, C) **then**
      newpath.remove(B)
    end if
  end for
end for
return newpath

E. Robot Vision

II. Design

A. System Structure

MENTION ALGORITHM COMPLEXITY! brief ROS description, callback based system, finite state automaton

B. Platform

Stuff about the pioneer—available sensors, some data about its size, specifications, our additions to it. Kinect specs. Include a picture of the robot with the kinect on it.

III. Experimentation

A. PRM

inflated map - show inflated map superimposed onto the original map Redo experiment for sampling methods. short, medium, long path length. Display image of map with one of the routes displayed and show the difference between the sampling methods. Find the optimum route by sampling a massive number of vertices on to the space and then finding a route using that—the flattened path is then the most optimal route, and we compare the other routes to this route for each experiment.

B. Vision

C. Exploration

IV. Discussion

A. Performance

B. Potential Improvements

C. Conclusions

References

[1] D. Kortenkamp, I. Nourbakhsh, and D. Hinkle, "The 1996 aaai mobile robot competition and exhibition," in *AI Magazine*, vol. 18, 1997.

[2] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Intelligent Robotics and Autonomous Agents Series, Mit Press, 2005.