COMP3411/9814: Artificial Intelligence

2d. Motion Planning

Alan Blair

School of Computer Science and Engineering

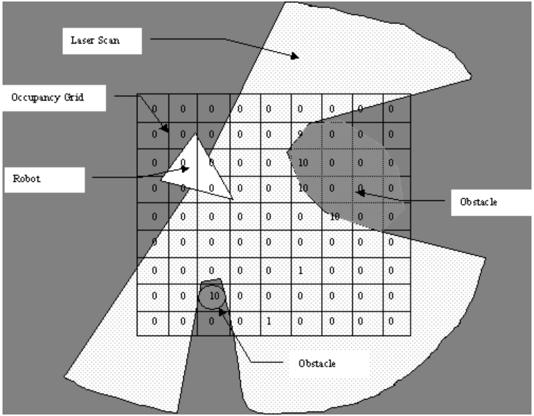
February 22, 2024

Motion Planning Approaches

- ➢ Partially Observable Environments (on-board sensors only)
  - → Occupancy Grid
  - → Potential Field
  - → Vector Field Histogram
- ➢ Fully Observable Environments (overhead cameras)
  - → Delaunay Triangulation
  - → Parameterized Cubic Splines

Robots

Occupancy Grid



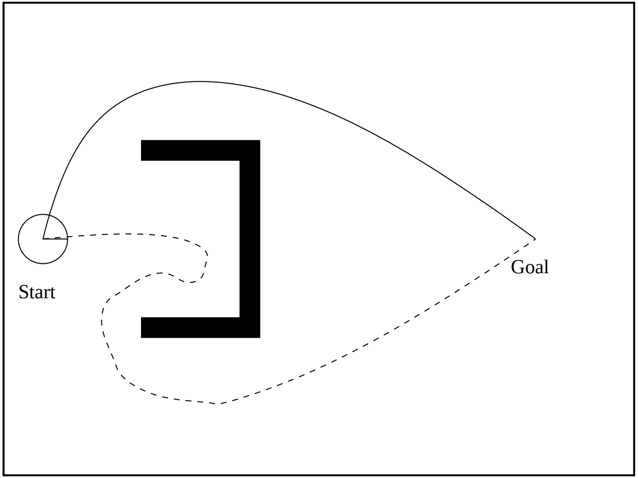
- ➛ divide environment into a Cartesian grid
- ➛ for each square in the grid, maintain an estimate of the probability of an obstacle in that square

Potential Field

Using an **occupancy grid** for its map, an agent can plan a path using a **Potential Field.**

- ➢ treat robot's configuration as a point in a potential field that combines attraction to the goal, and repulsion from obstacles
- ➢ very rapid computation, but can get stuck in local optima, thus failing to find a path

Problem - Local Optima



Vector Field Histogram

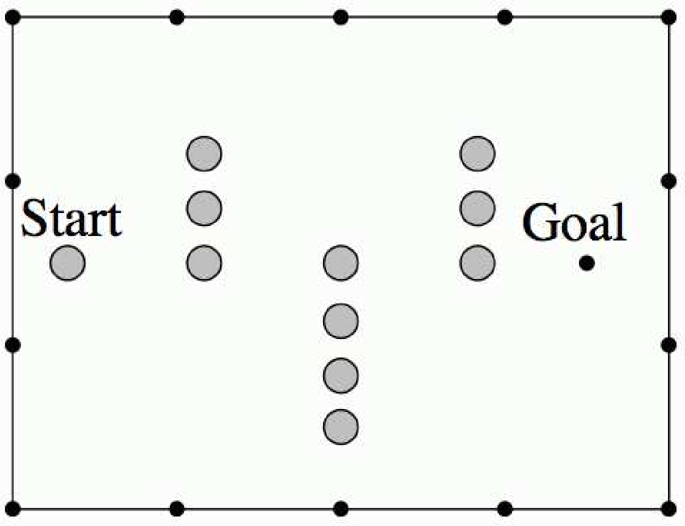
A more sophisticated approach -– less likely to get stuck in local optima -– is a **Vector Field Histogram**

- ➛ Cartesian histogram grid, continuously updated
- ➛ Polar histogram, based on current position/orientation of robot
- ➛ candidate valleys (contiguous sectors with low obstacle density)
- ➛ select candidate valleys, based on proximity to target direction

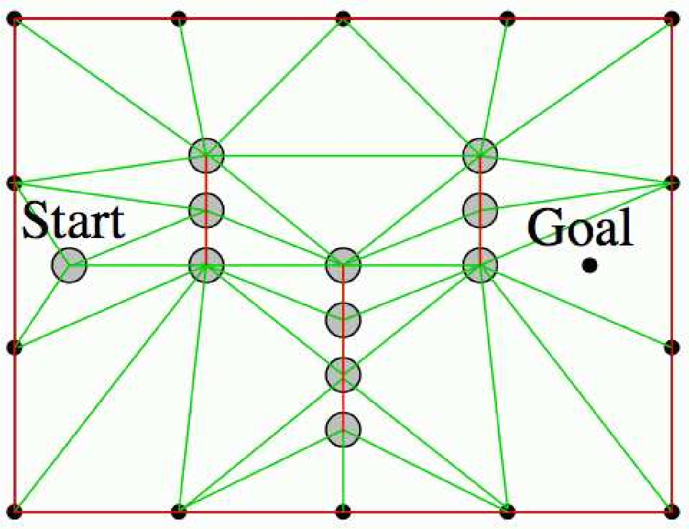
Delaunay Triangulation

- applicable in situations where the environment is well mapped by overhead cameras (museums, shopping centres, robocup small league)
- add line segments between (closest points of) obstacles, sorted according to length (shortest segments first)
- do not add any segment that crosses an existing segment

Delaunay Triangulation



Delaunay Triangulation



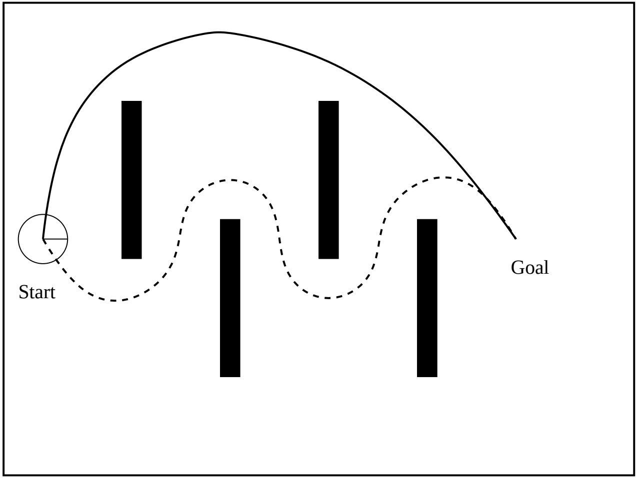
Voronoi Diagram

- ➛ dual of the Delaunay triangulation is called a **Voronoi diagram**
- ➛ prune arcs that are too small for the robot to traverse
- ➛ A*Search can then be applied on the resulting graph
- ➛ path can be converted to a trajectory for the robot

Minimizing Time instead of Distance

- For the problem of a soccer robot getting to the ball, or a wheeled robot navigating a maze, the "shortest distance" path might not be the same as the "shortest time" path.
- By speeding up and then slowing down, the robot could traverse a path with long straight stretches faster than a shorter path with lots of twists and turns.

Minimizing Time instead of Distance



How can we approach the problem of finding the “shortest time” path?

Optimal Trajectory Planning

Steps in the Algorithm:

- Delaunay Triangulation
- A*Search, using paths composed of Parametric Cubic Splines
- Smooth entire curve with Waypoint Tuning by Gradient Descent

Parameterized Cubic Splines

Assume each path segment is of the form

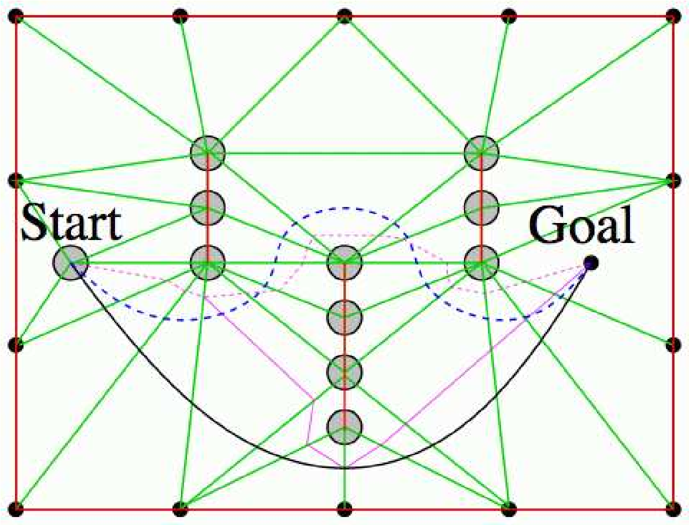
$$P(t) = \begin{pmatrix} P_x(t) \\ P_y(t) \end{pmatrix} = \begin{pmatrix} a_x \\ a_y \end{pmatrix} t^3 + \begin{pmatrix} b_x \\ b_y \end{pmatrix} t^2 + \begin{pmatrix} c_x \\ c_y \end{pmatrix} t + \begin{pmatrix} d_x \\ d_y \end{pmatrix}$$

Solve these equations for a specified position and velocity at the beginning ($t = 0$) and the end ($t = s$) of the segment. The traditional method was to set $s = 1$. We instead try to minimize s (total time for segment) while satisfying the kinematic constraints:

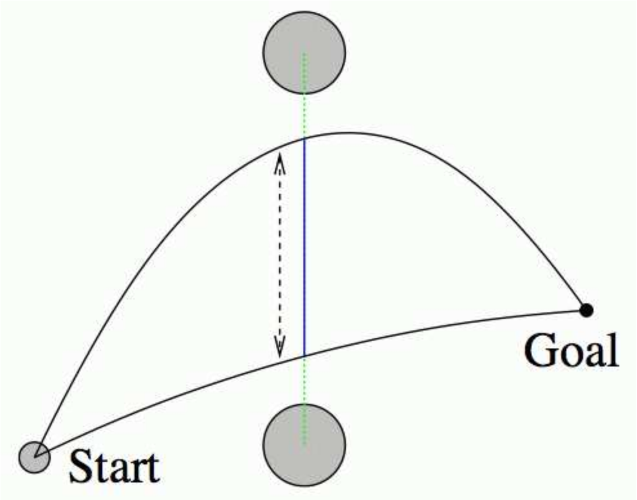
$$\left| \frac{P''(t)}{A} + \frac{P'(t)}{V} \right|^2 \leq 1, \qquad \text{for } 0 \leq t \leq s,$$

where A and V are the maximal acceleration and velocity of the robot.

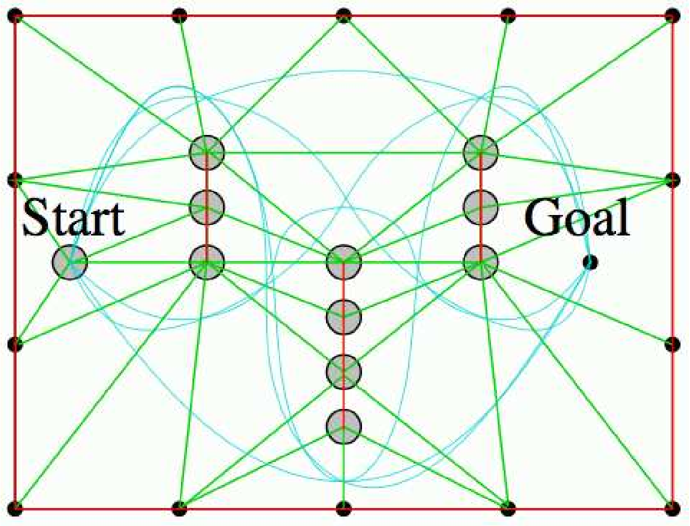
A*Search With and Without Smoothing



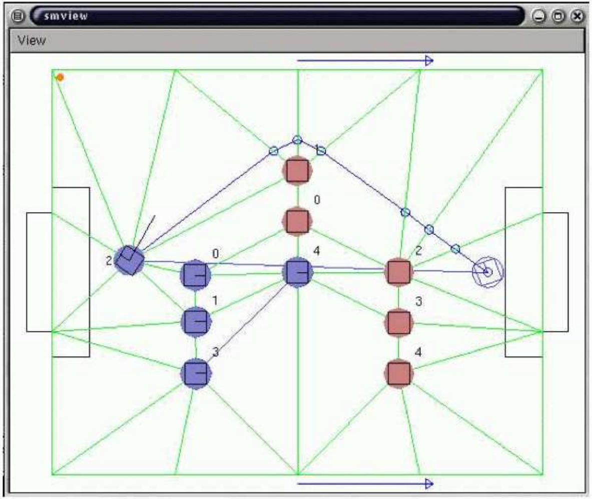
Waypoint Tuning by Gradient Descent



Various Paths Explored



Robocup



This system was successfully deployed in the Robocup F180 League.

Infinite Mario A*Search

References

- S. Thrun, W. Burgard & D. Fox, *Probabilistic Robotics*, MIT Press, 2005.
- J. Thomas, A. Blair & N. Barnes, "Towards an Efficient Optimal Trajectory Planner for Multiple Mobile Robots", *2003 International Conference on Robotics and Systems (IROS'03)*, 2291–2296.
- J. Togelius, S. Karakovskiy & R. Baumgarten, "The 2009 Mario AI Competition", *2010 IEEE Congress on Evolutionary Computation*.