Robot Programming

Localizing on a Distance Map

G. Grisetti

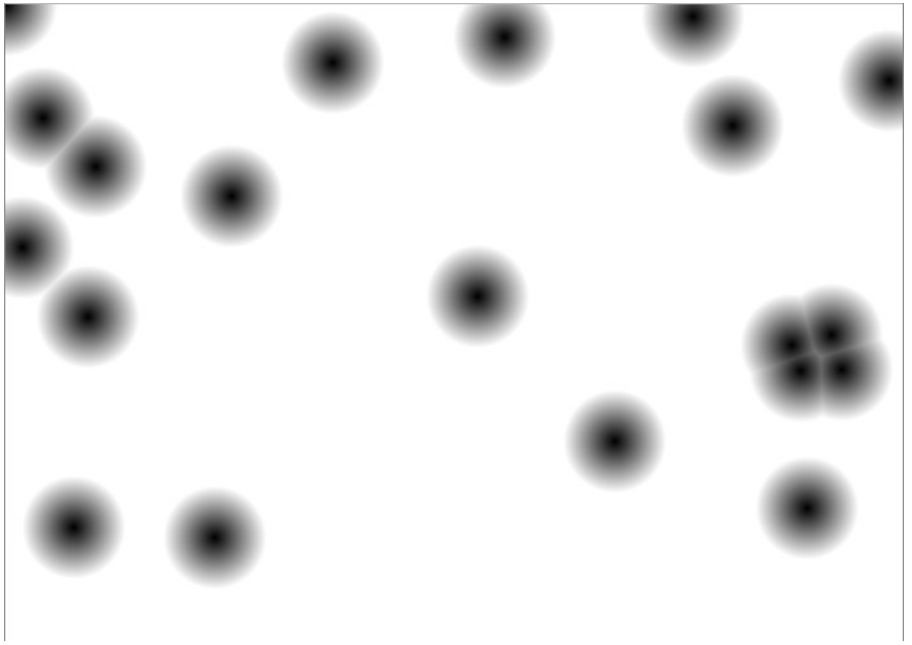
{grisetti}@diag.uniroma1.it

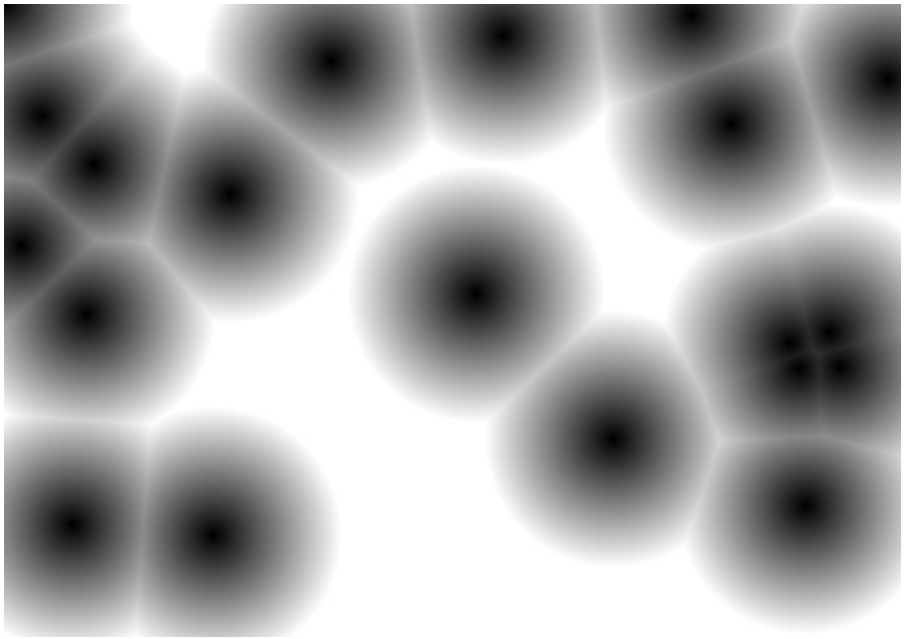
Department of Computer, Control, and Management Engineering
Sapienza University of Rome

DMap for Planning

The distance map is convenient to store the distances from the closest obstacle in a convenient manner

Can be expanded to account for a clearance

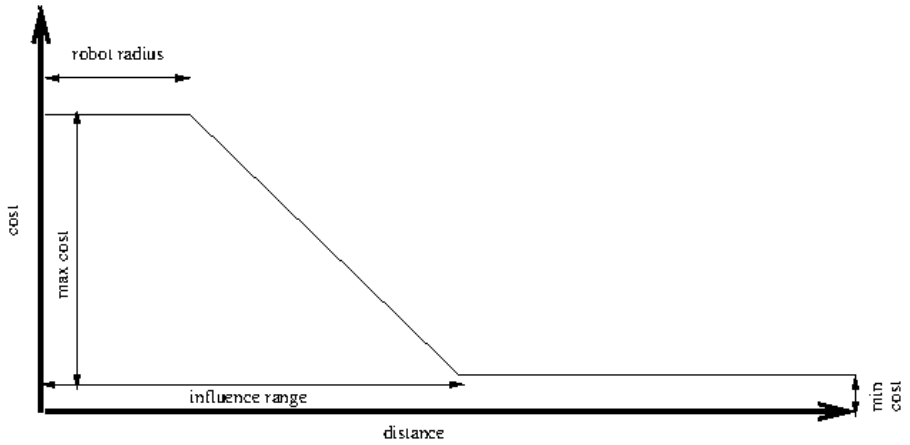


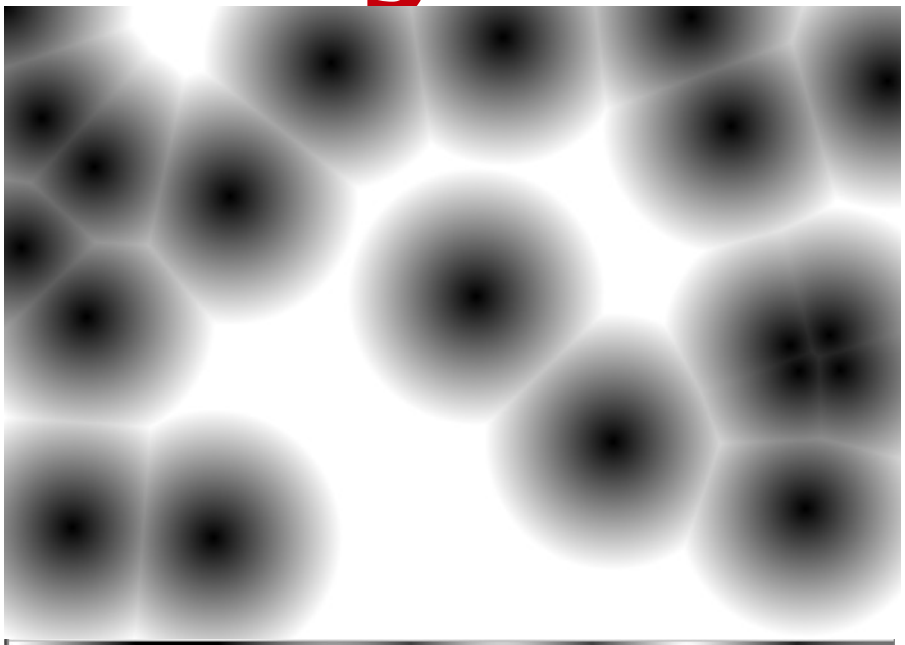


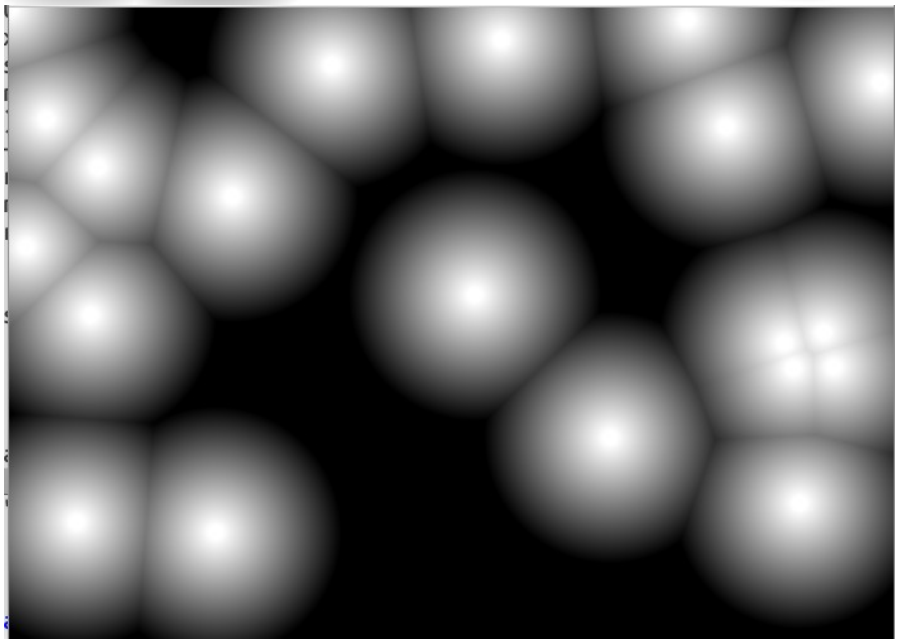
DMap for Planning

From the distance map we can compute the cost of being in a cell as a function of the distance

Max if d<r, then linearly decreasing







DMap for Planning

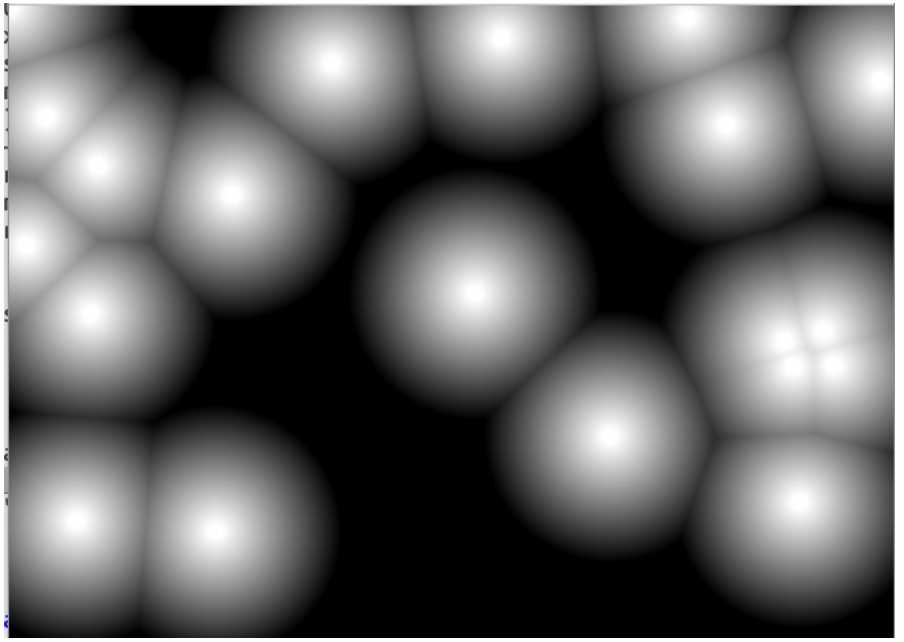
We can compute a path using an 8 connectivity, through a search algorithm.

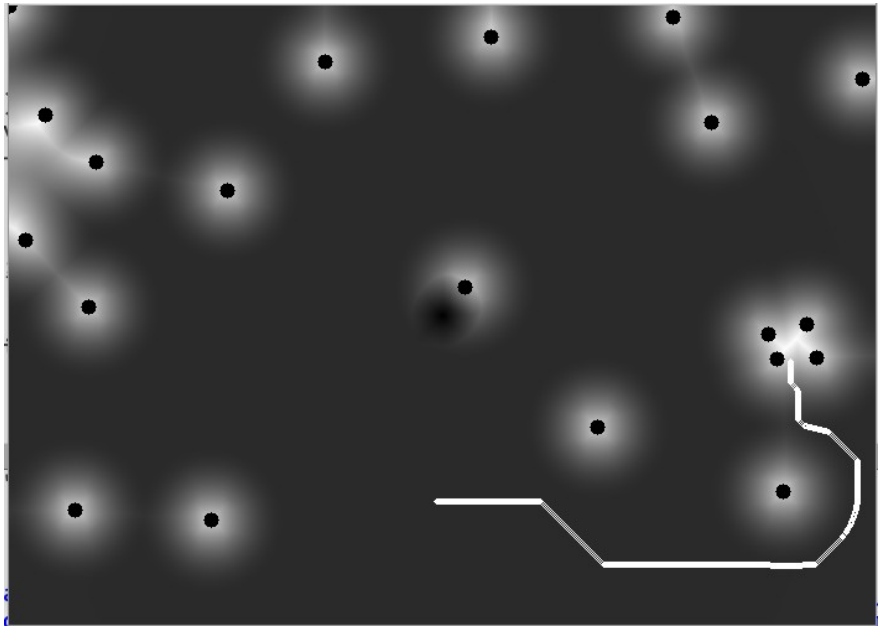
The cost of taking a move is proportional to

- the length of the motion (higher on the diagonal)
- The cost in the map

This gives us a **policy** that can be used to compute

- a path in linear time
- The next move in O(1)





Full Fledged Implementation

In the repo, you find a working Dmap Planner
The algorithm has a state **X** (the current estimate of the robot position)

- On startup the algorithm subscribes to
  - \map topic to get the occupancy grid on which to update distance and gradients
  - the current pose of the robot (through tf)
  - the \goal_pose msg that specifies the destination

Full Fledged Implementation

- Whenever it receives a pose update it produces a new path
- Whenever it receives goal msg it computes a new policy