DIJKSTRA'S AND FAST MARCHING METHOD FOR OPTIMAL PATH PLANNING

RYO TAKEI

ABSTRACT. The underlying PDE in solving the optimal path planning problem is the eikonal equation. Depending on the normed used, solutions to the eikonal equation, not surprisingly, differ.

INTRODUCTION

Optimal path planning is, essentially, a problem in optimal control. Hence, the formulation of the problem as a PDE exploits the *dynamic programming principle* (DPP):

$$V(x_0) = \inf_{y(\cdot)} \left[V(y(t+\Delta t)) + \int_t^{t+\Delta t} c(y(s))ds \right],$$

where V is the value function, c the running cost function, and $y(\cdot)$ the trajectory. With the domain D, we are given $T \subset D$, a set of targets where V is defined to be zero. The DPP can be made into a PDE by taking $\Delta t \to 0$:

$$\min_{\dot{y}} \nabla V(x) \cdot \dot{y} = c(x)$$

Under the constraint $||\dot{y}||_p \leq 1$ (or any constant), using the dual norm $|| \cdot ||_{p*}$, one arrives at the Eikonal equation:

$$\begin{aligned} ||\nabla V(x)||_{p*} &= c(x) \qquad \text{for } x \in D \backslash T \\ V(x) &= 0 \qquad \text{for } x \in \partial T \end{aligned}$$

For example, $|| \cdot ||_{1*} = || \cdot ||_{\infty}$, $|| \cdot ||_{\infty*} = || \cdot ||_1$ and the dual of $|| \cdot ||_2$ is itself.

DIJKSTRA'S ALGORITHM FOR NETWORKS

The optimal path problem considered on a (cartesian) grid amounts to solving a shortest path problem for discrete networks. Hence, the problem may be solved using Dijkstra's Algorithm: On a discrete grid, set $V = 0$ at the target set and $V = +\infty$ elsewhere. At each iteration, extract the grid point with the least value of V, say at x_j and update the value of $V(x_j)$ by,

$$V(x_j) \leftarrow \min_{x_k \in N_n(x_j)} \{w(x_k)c(x_j) + V(x_k)\}$$

where $N_n(x_j)$ are the neighbouring grid points of x_j, and $w(x_j)$ are the weights associated with each neighbouring grid point. Both N_n and w are dependent on the normed used for the eikonal equation. The norms with their respective neighbours and weights are shown in figure 1.

Date: August 11, 2006.

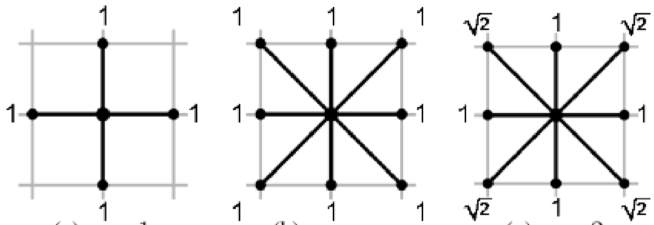

FIGURE 1. Norms (p) and their respective neighbours N_n and weights $w(x_j)$ of the neighbouring points.

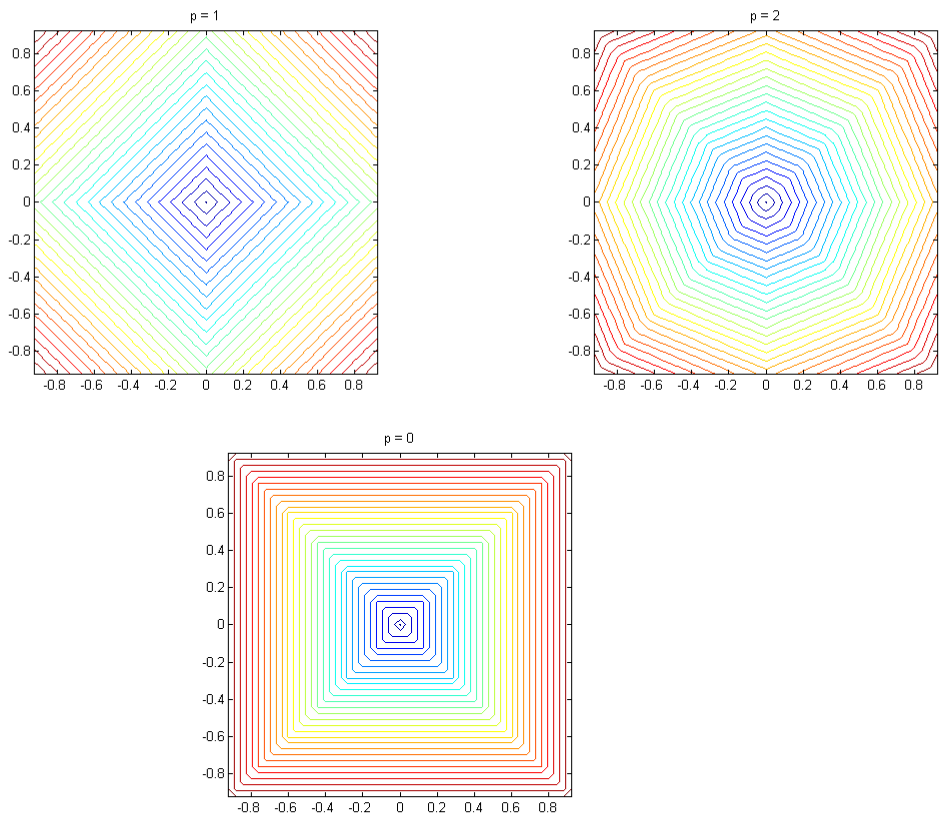

FIGURE 2. The contour lines of V computed by Dijktra's method using norms $p = 1, 2$ and ∞ (clockwise). Grid is size 51^2 for all cases.

The contour lines of V computed by Dijktra's method using norms $p = 1, 2$ and ∞ are shown in figure 2. For all cases, the target set T is the origin.

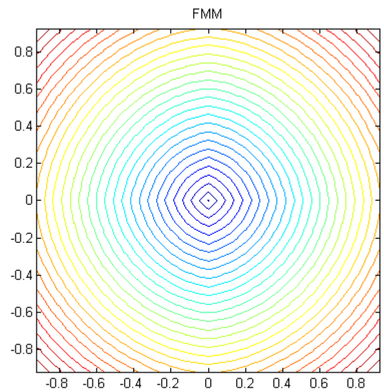

FIGURE 3. Norms (p) and their respective neighbours N_n and weights $w(x_j)$ of the neighbouring points. We see that the contour lines becomes circular as it moves away from the origin (the target T).

FAST MARCHING METHODS FOR EIKONAL EQUATIONS

Dijkstra's method, with the $p = 1$ stencil, for example, can be written as,

$$V_{i,j} = \min\{V_{i-1,j}, V_{i+1,j}, V_{i,j-1}, V_{i,j+1}\} + c_{i,j}$$

which is equivalent to:

$$\max\{D^{-x}V_{ij}, -D^{+x}V_{ij}, D^{-y}V_{ij}, -D^{+y}V_{ij}\} = c_{ij}.$$

Hence, Dijkstra's method with $p = 1$ is essentially solving the PDE:

$$\max\{|V_x|, |V_y|\} = c(x, y)$$

Likewise, for $p = 2$, Dijktra's method is solving the PDE:

$$\max_u\{\left|\frac{dV}{du}\right|\} = c(x, y)$$

where $u = (\cos\theta, \sin\theta)$ for $\theta = 0, \pi/4, \pi/2, \ldots, 7\pi/4$.

The *Fast Marching Method* (FMM), attempts to solve the underlying PDE,

$$\sqrt{V_x^2 + V_y^2} = c(x, y),$$

by applying the following discretization in the update step:

$$(\max\{D^{-x}V_{ij}, -D^{+x}V_{ij}, 0\}^2 + \max\{D^{-y}V_{ij}, -D^{+y}V_{ij}, 0\}^2)^{1/2} = c_{ij}$$

Solving the above equation for V_{ij} involves a quadratic equation:

$$V_{ij} = -(\alpha + \beta)/2 + (-(\alpha - \beta)^2 + 2c_{ij}^2)^{1/2}/2,$$

where $\alpha = -\min\{V_{i+1,j}, V_{i-1,j}\}$ and $\beta = -\min\{V_{i,j-1}, V_{i,j+1}\}$. For special cases when $V_{i+1,j} = V_{i-1,j} = +\infty$, we simply have $V_{ij} = c_{ij} - \beta$ and similarly when $V_{i,j-1} = V_{i,j+1} = +\infty$, we have $V_{ij} = c_{ij} - \alpha$.
The contour lines for V constructed using FMM is shown in figure 3. We see that the contour lines becomes circular as it moves away from the origin (the target T).

Sample Test Cases

Below are results of running Dijkstra's method for $p = 2$ and the FMM on a sample map with a given set of obstacles (where c is set to ∞). Optimal paths from various initial points are computed by a simple discrete analogue of gradient descent: simply move to the direction of minimum value in the neighbour points. Initial points for both cases were $(-0.6, -0.8)$, $(0.8, 0.2)$ and $(-0.4, 0.8)$. Results are shown in figure 4. The grid size was 81 by 81.
Results show better contour lines for FMM, while optimal trajectories computed were identical. This suggests that crude trajectory computing algorithms can give bad results even with a better value function V. More on this will be studied.
Remark: Ian Mitchell, in his paper, used this simple discrete analogue of gradient descent; why identical trajectories resulted in both Dijkstra and FMM is yet a question. Is the V found by FMM (I based mine on Sethian's book) different from those produced by Mitchell??

Department of Mathematics, Simon Fraser University
E-mail address: `rrtakei@sfu.ca`

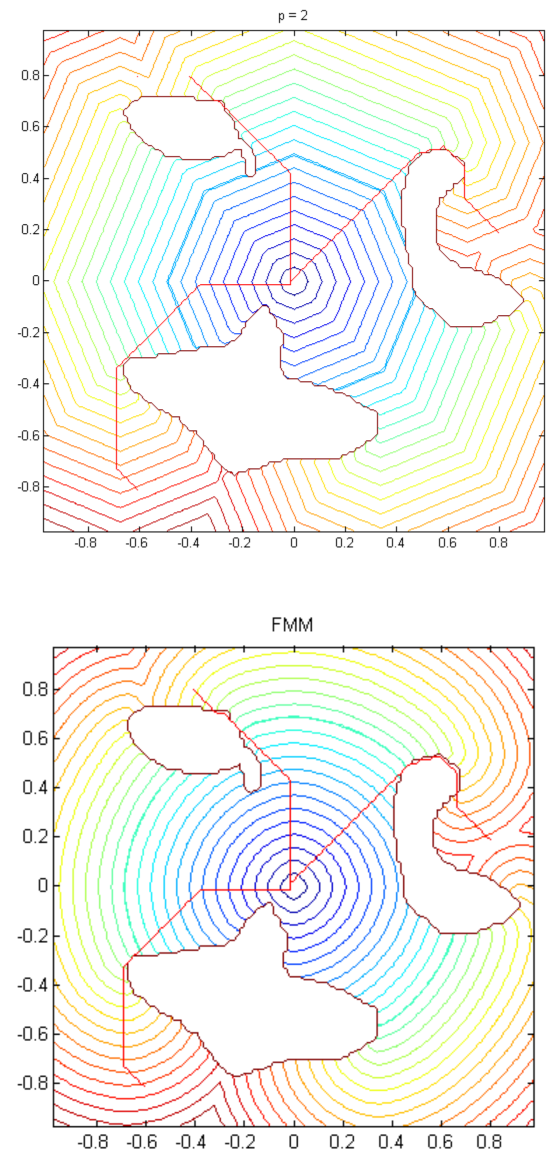

FIGURE 4. The contour lines of V computed by Dijktra's method ($p = 2$) and FMM for a sample map with obstacles. Optimal trajectories are shown from initial points $(-0.6, -0.8)$, $(0.8, 0.2)$ and $(-0.4, 0.8)$.