Tutorial 7

Fast Marching Methods

Guy Rosman

©2008
How to measure distances?

Given a domain $\Omega$ (for a moment assume that $\Omega \subset \mathbb{R}^2$) and a set of source points $X_0$, for each point $x \in \Omega$, measure its distance from $X_0$, $T(x)$.
Formally, the problem boils down to solving the eikonal equation (from Greek εικων = image).

\[
\|\nabla_\Omega T(x, y)\| = 1, \quad T(X_0) = 0.
\]
In optics and acoustics this equation describes the propagation of electromagnetic or pressure waves through some (non-homogenous) medium. It is responsible for the fact that the light (sound) traverses the shortest path between two points (Fermat’s principle) and as consequence it describes light refraction and Snell’s law.
Fermat’s principle - a simple example..
Fermat’s principle - a not-so-simple example..
Imagine that $\Omega$ is a uniformly distributed forest. At time $T = 0$, a bad guy sets fire simultaneously in all the “source points” $X_0$. The fire front advanced outwards from these initial points. Firemen register the time $T(x)$ at which the fire arrives to the location $x$. Once a point is touched by the fire, the trees burn out and the front never visits again the locations where it had already passed. The fast marching algorithm is a numerical tool, which simulates the described scenario.
The Fast Marching algorithm

A few observations when computing a solution to the Eikonal equation:

- Our boundary solution is also our zero level set - the set of source points
- The construction of the solution should be made monotonous, going from “closer” points to “farther” points - we are validating the triangle inequality on several stencils
The equation, as a solution to the minimum distance problem, is well defined in terms of narrow band around a level set.
The Fast Marching algorithm

Initialization

1. Set $T(X_0) = 0$ and mark the points from $X_0$ as Black.
2. Set the rest of the points $T(\Omega \setminus X_0) = \infty$ and mark them $X_0$ as Green.

Iteration

1. Green points adjacent to Black points become Red.
2. Red points are updated, i.e. their $T$ is estimated from the arrival times of the neighboring Black points.
3. The Red point with the smallest $T$ becomes Black.
4. The process reiterates until all points are Black.
Note that a point is updated only by adjacent points (4-neighbor adjacency) with smaller value of $T$, and we demand that the new $T$ value is larger than that of the updating points. A *Black* point has the $T$ value smaller than that of all non-*Black* points. Therefore, once a point becomes *Black*, it need not be updated anymore.

The maximum number of point updates is the number of neighbors on the grid, which is $O(N)$. 

![Diagram of grid with Black points](image)
The update step

When we say “update $x_3$ from $x_1, x_2$”, we mean:

Given two points $x_1, x_2$ with known front arrival times $T_1, T_2$, estimate the arrival time $T_3$ to the point $x_3$. 
Assume that the front is \textit{planar} (a good approximation for $T \gg h$), i.e. propagating from some plane $\hat{n} \cdot x + p = 0$.

The points $x_i$ are distant $T_i$ from the plane.

The knowledge of $T_1, T_2$ allows to determine the parameters of the plane $\hat{n}, p$. 
Assume w.l.o.g. that

\[ x_1 = (0, 0), \ x_2 = (\sqrt{2}h, 0), \ x_3 = \left( \frac{\sqrt{2}}{2}h, \frac{\sqrt{2}}{2}h \right) \]
We have:

\[ T_1 = \hat{n} \cdot x_1 + p = p \]
\[ T_2 = \hat{n} \cdot x_2 + p = \sqrt{2}hn_x + T_1 \]

from where

\[ n_x = \frac{T_2 - T_1}{\sqrt{2}h} \]

\[ n_y = \pm \sqrt{1 - n_x^2} = \pm \frac{1}{\sqrt{2h}} \sqrt{2h^2 - (T_1 - T_2)^2}. \]
Since $x_3$ is “above” $x_1$ and $x_2$ w.r.t. the $y$-axis, the solution $n_y = -\frac{1}{\sqrt{2}h} \sqrt{2h^2 - (T_1 - T_2)^2}$ corresponds to the source, from which the front arrives to $x_3$ before it arrives to $x_1$ and $x_2$, yielding $T_3 < T_1, T_2$. We discard this solution.

$$T_3 = \hat{n} \cdot x_3 + p = \frac{\sqrt{2}}{2} hn_x + \frac{\sqrt{2}}{2} hn_y + T_1$$
$$= \frac{1}{2} (T_2 - T_1) + \frac{1}{2} \sqrt{2h^2 - (T_2 - T_1)^2} + T_1$$
$$= \frac{1}{2} (T_2 + T_1) + \frac{1}{2} \sqrt{2h^2 - (T_2 - T_1)^2}$$
$$= \frac{1}{2} \left( T_1 + T_2 + \sqrt{2h^2 - (T_2 - T_1)^2} \right).$$
If $|T_1 - T_2| > h$, the wavefront propagation direction $n$ comes outside of the triangle $x_1x_2x_3$ and arrives to $x_3$ before arriving to $x_1$ or $x_2$.

This can be seen from the algebra. Assume $T_1 > T_2$ and $T_1 > T_2 + h$,

$$T_3 = \frac{1}{2} \left( T_1 + T_2 \pm \sqrt{2h^2 - (T_2 - T_1)^2} \right) < \frac{1}{2} (T_1 + T_2 + h) < \frac{1}{2} (2T_2 + 2h) = T_2 + h < T_1.$$  

In this case, we have no choice but to update $x_3$ according to

$$T_3 = \min\{T_1, T_2\} + h.$$
Extension to the weighted Euclidean case

When the wavefront has non-uniform propagation velocity $V(x, y)$, a *weighted* distance map is computed by solving the eikonal equation

$$
\| V(x, y) \nabla T(x, y) \| = 1
$$

or

$$
\| \nabla T(x, y) \| = F(x, y),
$$

where $F(x, y) = V^{-1}(x, y) > 0$ is the *slowness* field.
This equation can be reformulated as

\[ \| \nabla g T(x, y) \| = 1, \]

where the front arrival time is measured according to the weighted Euclidean metric, \( dt = \sqrt{dx^2 + dy^2}F(x, y) \).

In the weighted case, the time gained by traversing the distance \( h \) when updating \( x_3 \) from the neighboring points \( x_1, x_2 \) is not \( h \) but \( hF(x, y) \). Hence,

\[
T_3 = \frac{1}{2} \left( T_1 + T_2 + \sqrt{2F^2h^2 - (T_2 - T_1)^2} \right),
\]

valid for \( |T_1 - T_2| \leq hF \).
Extension to (acute) triangulated manifolds

- The eikonal equation can be solved on general surfaces, represented as triangulated meshes (polyhedral or piecewise-linear approximation).
- The algorithm is very similar, yet now each vertex on the mesh is updated from the triangles built upon it.
- We first deal with triangles with acute angle.
We are looking for \( t = EC \), where

\[
t > u
\]

Note:

\[
CD = b \frac{t - u}{t}
\]

Using the law of cosines:

\[
BD^2 = CD^2 + a^2 - 2aCD\cos\theta
\]

Using the law of sines:

\[
\frac{\sin\phi}{CD} = \frac{\sin\theta}{BD}
\]
Looking at the triangle $CBG$, we have

$$h = a \sin \phi = a \frac{CD}{BD} \sin \theta$$

$$= \frac{aCD \sin \theta}{\sqrt{CD^2 + a^2 - 2aCD \cos \theta}}$$

setting $h$ in the equation

$$\frac{t - u}{h} = F$$

We obtain:

$$\frac{(t - u)\sqrt{CD^2 + a^2 - 2aCD \cos \theta}}{aCD \sin \theta} = F \rightarrow$$

$$\frac{(t - u)\sqrt{CD^2 + a^2 - 2aCD \cos \theta}}{ab \frac{(t - u)}{t} \sin \theta} = F \rightarrow$$
\[
\sqrt{CD^2 + a^2 - 2aCD\cos\theta} \quad \frac{ab^1}{t} \sin\theta = F \rightarrow \\
\sqrt{\left(\frac{b(t-u)}{t}\right)^2 + a^2 - 2a \left(\frac{b(t-u)}{t}\right) \cos\theta} = Fb\frac{1}{t} \sin\theta \rightarrow \\
\left(\frac{b(t-u)}{t}\right)^2 + a^2 - 2a \left(\frac{b(t-u)}{t}\right) \cos\theta = F^2 a^2 b^2 \frac{1}{t^2} \sin^2\theta \rightarrow 
\]
\[(b(t - u))^2 + a^2 t^2 - 2tab(t - u) \cos \theta = F^2 a^2 b^2 \sin^2 \theta \rightarrow\]

\[b^2 \left( t^2 - 2tu + u^2 \right) + a^2 t^2 - 2 \left( t^2 - tu \right) ab \cos \theta = F^2 a^2 b^2 \sin^2 \theta \rightarrow\]

\[t^2 \left( a^2 + b^2 - 2ab \cos \theta \right) + t \left( 2uab \cos \theta - 2ub^2 \right)\]

\[+ b^2 \left( u^2 - F^2 a^2 \sin^2 \theta \right) = 0\]
Note that \( t > u \) according to our assumptions, and \( CD \) should stay within the segment \( CA \). By changing the direction of \( h \) we obtain:

\[
acos\theta < \frac{b(t - u)}{t} < \frac{a}{cos\theta}
\]
The discussion so far has been for non-obtuse triangulations. If obtuse triangles exist, they can be “split” into 2 by adding an extra edge (and 2 virtual triangles) to the triangulation graph (which now becomes non-planar, but still works for our purposes)

- **Obtuse angle and splitting section**
- **Triangulated surface patch**
- **Unfolded triangulated patch**

![Diagram of obtuse angle and splitting section]

![Diagram of triangulated surface patch]

![Diagram of unfolded triangulated patch]