

A Tutorial on
**Level Set and Implicit Models for
Solid Processing and Modeling**

Michael Y. Wang
Qi Xia

Department of Automation & Computer-
Aided Engineering
The Chinese University of Hong Kong
www.acae.cuhk.edu.hk/~yuwang

Contents

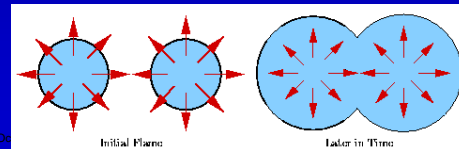
- Level set method
 - The concept
 - Discrete level set computational schemes
- Level set method for solid optimization
 - Front propagation and speed functions
 - Structural optimization examples
- Implicit surface models with radial basis functions
 - Implicit representation
 - Radial basis functions (RBFs)
 - Implicit surface reconstruction from scattered data
 - Implicit surface processing

Level Set Method

- Concept of Level Set Method
- Level Set and External Velocity
 - hyperbolic PDE
 - upwind differencing and Hamilton-Jacobi schemes
- Level Set and Mean Curvature Velocity
 - parabolic PDE
 - central differencing
- Level Set and Constant velocity
 - hyperbolic PDE and its numerical formula
- All the Velocities
- Signed Distance Function Reinitialization

Level Set Method

- Contour evolution method due to:
J. Sethian and S. Osher, 1988
- Advantages:
 - Eulerian surface evolution
 - with changes in topology

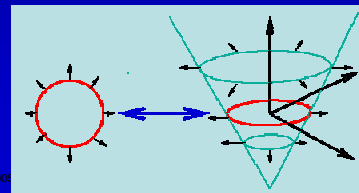


The Concept

- The level set approach:
 - Define problem in 1 higher dimension
 - Define level set function $z = \phi(x, y, t = 0)$ where the (x, y) plane contains the contour, and $z =$ signed Euclidean distance transform value (negative for inside closed contour, positive for outside contour)

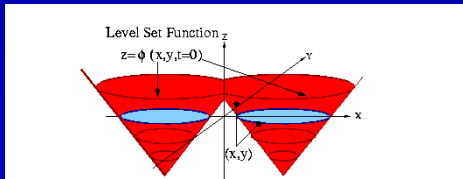
How to Move the Contour?

- Move the level set function, $\phi(x, y, t)$, so that it rises, falls, expands, etc.
- Contour = cross section at $z = 0$, i.e., $\{(x, y) \mid \phi(x, y, t) = 0\}$



Level Set Surface

- The zero level set (in blue) at one point in time as a slice of the level set surface (in red)



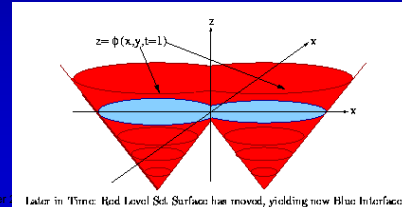
The Level Set Surface (in red) plots the distance from each point (x, y) to the Interface (in blue)

19 Oct

7

Level Set Surface

- Later in time the level set surface (red) has moved and the new zero level set (blue) defines the new contour

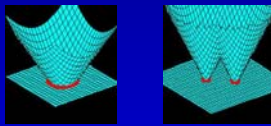


Later in Time: Red Level Set Surface has moved, yielding new Blue Interface

19 October

8

Level Set Surface



19 October 2005

Michael Y. Wang

9

How to Move the Level Set Surface?

- Define a velocity field, F , that specifies how contour points move in time
 - Based on application-specific physics such as time, position, normal, curvature, image gradient magnitude
- Build an initial value for the level set function, $\phi(x, y, t=0)$, based on the initial contour position
- Adjust ϕ over time; contour at time t defined by $\phi(x(t), y(t), t) = 0$

19 October 2005

Michael Y. Wang

10

Level Set Formulation

- Constraint: level set value of a point on the contour with motion $\mathbf{x}(t)$ must always be 0

$$\phi(\mathbf{x}(t), t) = 0$$

- By the chain rule

$$\phi_t + \nabla \phi(\mathbf{x}(t), t) \cdot \mathbf{x}'(t) = 0$$

- Since F supplies the speed in the outward normal direction

$$\mathbf{x}'(t) \cdot \mathbf{n} = F, \text{ where } \mathbf{n} = \nabla \phi / |\nabla \phi|$$

- Hence evolution equation for ϕ is

$$\phi_t + F|\nabla \phi| = 0$$

19 October 2005

Michael Y. Wang

11

Hamilton-Jacobi equation

$$\frac{\partial \Phi}{\partial t} + \bar{F} \cdot \nabla \Phi = 0$$

$$\frac{\partial \Phi}{\partial t} + F \left(\left(\frac{\partial \Phi}{\partial x} \right)^2 + \left(\frac{\partial \Phi}{\partial y} \right)^2 \right)^{1/2} = 0$$

19 October 2005

Michael Y. Wang

12

Speed Function

$$F(k) = F_0 + F_1(k) = (I - \varepsilon k)$$

$$F(k) = k_1(x, y) * (I - \varepsilon k)$$

$$k_1 = \frac{1}{1 + |\nabla G_{\sigma} * I(x, y)|}$$

$$k_1 = e^{-|\nabla G_{\sigma} * I(x, y)|}$$

19 October 2005

Michael Y. Wang

13

Example: Shape Simplification

- $F = 1 - 0.1\kappa$ where κ is the curvature at each contour point



19 October 2005

14

- Concept of Level Set Method
- Level Set and External Velocity
 - hyperbolic PDE
 - upwind differencing and Hamilton-Jacobi schemes
- Level Set and Mean Curvature Velocity
 - parabolic PDE
 - central differencing
- Level Set and Constant velocity
 - hyperbolic PDE and its numerical formula
- All the Velocities at One Time
- Signed Distance Function Reinitialization

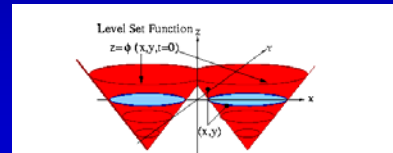
19 October 2005

Michael Y. Wang

15

Level Set Methods

- The red surface $\phi(x, t)$ is called the level set function.
- The blue interface $\phi(x, t) = 0$ is called the zero level set.
- Level Set is a systematic method to change the height of the surface $\phi(x, t)$ to match the evolution of the interface.



19 October 2005

The Level Set Surface (in red) plots the distance from each point (x, y) to the Interface (in blue)

16

Level Set Methods

- Suppose $x(t)$ is a particle trajectory on the interface $\Gamma(t)$ moving with velocity $\vec{V} = \dot{x}(t)$. By definition $\phi(x(t), t) = 0$, differentiating with respect to t , we get

$$\phi_t + \vec{V} \cdot \nabla \phi = 0$$

- The equation is called **Level Set Equation**. And one should be aware that the equation is **only defined on interface**. The level set method is to **extend Level Set Equation to be valid throughout the domain and pick up the zero level set as the interface at all later time**.

19 October 2005

Michael Y. Wang

17

Level Set Methods

- Many geometric properties of Γ have simple expression in terms of ϕ , because ϕ contains local information which allows implicit differentiation of Γ , for example,

$$N = \nabla \phi / |\nabla \phi|, \quad \kappa = \nabla \cdot N$$

- These formulas can be evaluated **everywhere** ϕ is known, as well as on Γ . At a point away from Γ , they give the geometry of the level set passing through x .

19 October 2005

Michael Y. Wang

18

Level Set Methods

- The level set equation move **every level set** of ϕ with the extended velocity \mathbf{V} or \mathbf{F} , and in particular move the zero set with the correct velocity.
- This approach to moving interfaces **embeds** the topology in ϕ rather than Γ , allowing merging, breaking, and other topological changes to be handled automatically. We pay the price of going up one dimension.
- Either \mathbf{V} or \mathbf{F} must be extended to a function on the whole space, but the extension can be almost completely arbitrary away from Γ .

19 October 2005

Michael Y. Wang

19

Level Set and External Velocity Field

- When \vec{V} is given from **external physics**, such as advection, the level equation contains only **first order derivative** of ϕ , i.e. it is **hyperbolic**.

$$\phi_t + \vec{V} \cdot \nabla \phi = 0$$

where $\vec{V} = (u, v)$.

- By projecting \vec{V} to the direction \vec{n} normal to the interface, we get

$$\phi_t + V_n |\nabla \phi| = 0$$

- For External Velocity (Passive Transport), V is already defined on whole space and is a natural extension of zero level set velocity.

19 October 2005

Michael Y. Wang

20

Level Set and External Velocity Field

- The simplest numerical method to solve the hyperbolic level set equation is by a combination of **forward Euler time discretization** and **upwind spatial differencing**.

- Its numerical formula is

$$\phi_{ij}^{n+1} = \phi_{ij}^n - \Delta t [\max(u_{ij}^n, 0) D_{ij}^{-x} + \min(u_{ij}^n, 0) D_{ij}^{+x} + \max(v_{ij}^n, 0) D_{ij}^{-y} + \min(v_{ij}^n, 0) D_{ij}^{+y}]$$

where (u_{ij}^n, v_{ij}^n) is the external velocity at time n and at point (i, j) , and Δt must satisfy CFL (Courant-Friedrichs-Lewy) condition.

$$\Delta t \max \left\{ \frac{|u|}{\Delta x} + \frac{|v|}{\Delta y} \right\} < \alpha, \quad 0 < \alpha < 1$$

19 October 2005

Michael Y. Wang

21

Hamilton-Jacobi equation

- Equation of the form

$$\phi_t + H(\phi_x, \phi_y) = 0$$

is called Hamilton-Jacobi equation.

- Hamilton-Jacobi equations depend on (at most) the **first order derivatives of ϕ** , and these equations are **hyperbolic**.

- For equation $\phi_t + \vec{V} \cdot \nabla \phi = 0$, $H = \vec{V} \cdot \nabla \phi$.

- For equation $\phi_t + V_n |\nabla \phi| = 0$, $H = V_n |\nabla \phi|$, and it is required that there is **no high order derivatives of ϕ with respect to x or y in the expression of V_n** .

19 October 2005

Michael Y. Wang

22

Hamilton-Jacobi equation

- A forward Euler time discretization of a Hamilton-Jacobi equation can be written as

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} + \hat{H}^n(\phi_x^-, \phi_x^+, \phi_y^-, \phi_y^+) = 0$$

where the $\hat{H}^n(\phi_x^-, \phi_x^+, \phi_y^-, \phi_y^+)$ is the numerical approximation of $H(\phi_x, \phi_y)$.

- The CFL (Courant-Friedrichs-Lewy) condition for the above equation is

$$\Delta t \max \left\{ \frac{|H_1|}{\Delta x} + \frac{|H_2|}{\Delta y} \right\} < 1$$

19 October 2005

Michael Y. Wang

23

Hamilton-Jacobi equation

- H_1, H_2 are partial derivatives of H with respect to ϕ_x, ϕ_y .

- For $H = \vec{V} \cdot \nabla \phi$, where $\vec{V} = (u, v)$

$$H_1 = u, \quad H_2 = v$$

- For $H = V_n |\nabla \phi|$,

$$H_1 = V_n \phi_x / |\nabla \phi|, \quad H_2 = V_n \phi_y / |\nabla \phi|$$

19 October 2005

Michael Y. Wang

24

Hamilton-Jacobi equation

- The Lax-Friedrichs Scheme to approximate H is

$$\hat{H} = H\left(\frac{\phi_x^- + \phi_x^+}{2}, \frac{\phi_y^- + \phi_y^+}{2}\right) - \alpha^x \left(\frac{\phi_x^+ - \phi_x^-}{2}\right) - \alpha^y \left(\frac{\phi_y^+ - \phi_y^-}{2}\right)$$

Central differencing Artificial dissipation

where $\alpha^x = \max |H_1(\phi_x, \phi_y)|$, $\alpha^y = \max |H_2(\phi_x, \phi_y)|$

- For $H_1 = u, H_2 = v$, α^x, α^y are independent of ϕ_x, ϕ_y , α^x, α^y can be set to the maximum values of $|u|, |v|$.
- For $H(\phi) = V_n |\nabla \phi|$, $H_1 = V_n \phi_x / |\nabla \phi|, H_2 = V_n \phi_y / |\nabla \phi|$, so α^x, α^y can be evaluated by maximum $|V_n \phi_x|$ and minimum $|\phi_y|$, or simply set to the maximum value of $|V_n|$ in the domain.

19 October 2005

Michael Y. Wang

25

Hamilton-Jacobi equation

- Several other schemes that similar to LF scheme were devised, such as SLF (stencil Lax-Friedrichs), LLF (Local Lax-Friedrichs), LLLF (Local Local Lax-Friedrichs). Each method has its own search region.

In practice, LLF seems better than any of the other options. LF and SLF are usually too dissipative, while LLLF is usually not dissipative enough to overcome the problems introduced by using the centrally averaged approximation to ϕ_x, ϕ_y in evaluating H .

19 October 2005

Michael Y. Wang

26

Hamilton-Jacobi equation

- Choosing the appropriate amount of artificial dissipation to add to the centrally evaluated H is tricky.
- Therefore, it is often desirable to use upwind based methods with built-in artificial dissipation.
- Upwind discretizations have numerical truncation errors that serve the same purpose as artificial dissipation. First order accurate upwinding has an intrinsic $O(\Delta x)$ artificial viscosity. Higher order accurate upwind methods have intrinsic artificial viscosity with magnitude $O((\Delta x)^r)$, where r is the order of accuracy of the method.

19 October 2005

Michael Y. Wang

27

Hamilton-Jacobi equation

- Roe-Fix Scheme

$$\hat{H} = H(\phi_x^*, \phi_y^*) - \alpha^x \left(\frac{\phi_x^+ - \phi_x^-}{2}\right) - \alpha^y \left(\frac{\phi_y^+ - \phi_y^-}{2}\right)$$

- Use Roe's upwind method where sonic points does not form, and add an LLF entropy correction at sonic points where entropy-violating expansion shocks might form (The added dissipation from the LLF entropy correction forces the expansion shocks to develop into continuous rarefaction waves)

One can compute ϕ_x^\pm, ϕ_y^\pm using the first-order accurate forward and backward difference formulas and use these cheaper approximations to decide whether or not upwinding or LLF will be used. After decision, the higher-order accurate HJ WENO or HJ ENO method can be used to compute the necessary value of ϕ_x^\pm, ϕ_y^\pm .

Hamilton-Jacobi equation

- Godunov's Scheme

$$\hat{H} = \text{ext}_x \text{ext}_y H(\phi_x, \phi_y)$$

- For every point,
 - If $\phi_x^- < \phi_x^+$, then $\text{ext}_x H$ takes on the minimum value of H for ϕ_x^- or ϕ_x^+ .
 - If $\phi_x^- > \phi_x^+$, then $\text{ext}_x H$ takes on the maximum value of H for ϕ_x^- or ϕ_x^+ .
 - If $\phi_x^- = \phi_x^+$, then $\text{ext}_x H$ simply plugs $\phi_x^- (= \phi_x^+)$ into H for ϕ_x .
- $\text{ext}_y H$ is defined similarly to $\text{ext}_x H$.

19 October 2005

Michael Y. Wang

29

Hamilton-Jacobi equation

- Godunov's Scheme

$$\hat{H} = \text{ext}_x \text{ext}_y H(\phi_x, \phi_y)$$

- In our level set equation, H is separable with

$$\text{ext}_x \text{ext}_y H = \text{ext}_x (u \phi_x) + \text{ext}_y (v \phi_y)$$

- In this special case, Godunov's Scheme is identical to simple upwind differencing, that is,

$$\text{if } u > 0, \text{ use } \phi_x^-$$

$$\text{if } u < 0, \text{ use } \phi_x^+$$

$$\text{if } u = 0, \text{ set } u \phi_x = 0$$

19 October 2005

Michael Y. Wang

30

Level Set and Mean Curvature Velocity

- **Motion by mean curvature:** interface moves in the normal direction with a velocity proportional to its curvature:

$$\begin{cases} \phi_t + V_n |\nabla \phi| = 0 \\ V_n = -b\kappa \end{cases} \Rightarrow \phi_t = b\kappa |\nabla \phi|$$

where κ is mean curvature.

- When ϕ is signed distance function, i.e. $|\nabla \phi| = 1$, the level set equation can be **simplified** as

$$\phi_t = b\Delta \phi$$

- Since there are **second order derivative** in the above PDE, the PDE is **parabolic**.

19 October 2005

Michael Y. Wang

31

Level Set and Mean Curvature Velocity

- Using forward Euler time discretization, the numerical level set formula is

$$\phi_{ij}^{n+1} = \phi_{ij}^n + \Delta t \cdot b[(D^{-x} D^{+x})_{ij}^n + (D^{-y} D^{+y})_{ij}^n]$$

- where the spatial difference can be approximated by **central differencing**, i.e.

$$(D^{-x} D^{+x})_{ij}^n = \frac{\phi_{(i+1)j}^n - 2\phi_{ij}^n + \phi_{(i-1)j}^n}{\Delta x^2}$$

$$(D^{-y} D^{+y})_{ij}^n = \frac{\phi_{i(j+1)}^n - 2\phi_{ij}^n + \phi_{i(j-1)}^n}{\Delta y^2}$$

19 October 2005

Michael Y. Wang

32

Level Set and Mean Curvature Velocity

- After one time step, ϕ may not stay a signed distance function, so before we reinitialize ϕ , the simplified level equation can not be used.

- Central differencing of $\Delta \phi$ combined with a forward Euler time discretization requires a time-step restriction of

$$\Delta t \max\left\{\frac{2b}{(\Delta x)^2} + \frac{2b}{(\Delta y)^2}\right\} < \alpha, \quad 0 < \alpha < 1$$

- here $\Delta t = O((\Delta x)^2)$, Which is significantly **more stringent** than in the hyperbolic case, where $\Delta t = O(\Delta x)$.

19 October 2005

Michael Y. Wang

33

Level Set and Constant Velocity

- When interface moves in the normal direction with a constant velocity at every point on the interface, the level set equation is

$$\phi_t + a|\nabla \phi| = 0$$

- The above equation can be expanded as

$$\phi_t + \left(\frac{a\phi_x}{|\nabla \phi|}, \frac{a\phi_y}{|\nabla \phi|}, \frac{a\phi_z}{|\nabla \phi|}\right) \cdot \nabla \phi = 0$$

- The first spatial term is $\frac{a\phi_x}{|\nabla \phi|}$, the sign of which can be used to decide whether ϕ_x^- or ϕ_x^+ should be used to approximate ϕ_x .

19 October 2005

Michael Y. Wang

34

Level Set and Constant Velocity

- When ϕ_x^- and ϕ_x^+ have the same sign, upwinding can be used.

- When ϕ_x^- and ϕ_x^+ have different sign, simple upwinding breaks down, and we must turn to Hamilton-Jacobi schemes.

- RF scheme treats the ambiguity associated with upwinding near sonic points by using central differencing plus some artificial viscosity. However, it too resorts to artificial dissipation in the vicinity of sonic points.

- In order to avoid the addition of artificial dissipation, one must use the **Godunov scheme**.

19 October 2005

Michael Y. Wang

35

Level Set and Constant Normal Velocity

- The numerical formula for equation

$$\phi_t + a|\nabla \phi| = 0$$

is

$$\phi_{ij}^{n+1} = \phi_{ij}^n - \Delta t[\max(a,0)\nabla^+ + \min(a,0)\nabla^-]$$

$$\nabla^+ = [\max(D_j^{-x}, 0)^2 + \min(D_j^{+x}, 0)^2 + \max(D_j^{-y}, 0)^2 + \min(D_j^{+y}, 0)^2]^{1/2}$$

$$\nabla^- = [\min(D_j^{-x}, 0)^2 + \max(D_j^{+x}, 0)^2 + \min(D_j^{-y}, 0)^2 + \max(D_j^{+y}, 0)^2]^{1/2}$$

19 October 2005

Michael Y. Wang

36

All Velocities

- When we put all the velocity into level set equation, we get

$$\phi_t + \vec{V} \cdot \nabla \phi + a|\nabla \phi| = b\kappa|\nabla \phi|$$

this equation combines an external velocity with constant velocity and motion by mean curvature.

- The **parabolic term** on the right-hand side can be independently discretized with **central differencing**.
- The **hyperbolic part** of this equation consist of two terms, which are discretized by **HJ schemes**.

19 October 2005

Michael Y. Wang

37

All Velocities

- The numerical formula for the combined equation is

$$\begin{aligned} \phi_j^{n+1} = \phi_j^n - \Delta t \{ & [\max(a,0)\nabla^+ + \min(a,0)\nabla^-] \\ & + [\max(u_j^n,0)D_j^{+x} + \min(u_j^n,0)D_j^{+x} \\ & + \max(v_j^n,0)D_j^{+y} + \min(v_j^n,0)D_j^{+y}] \\ & - [b(D^{-x}D^{+x})_j^n + (D^{-y}D^{+y})_j^n] \} \end{aligned}$$

$$\nabla^+ = [\max(D_j^{+x},0)^2 + \min(D_j^{+x},0)^2 + \max(D_j^{+y},0)^2 + \min(D_j^{+y},0)^2]^{1/2}$$

$$\nabla^- = [\min(D_j^{-x},0)^2 + \max(D_j^{-x},0)^2 + \min(D_j^{-y},0)^2 + \max(D_j^{-y},0)^2]^{1/2}$$

- The CFL condition for equations that contain both hyperbolic and parabolic terms is given by

$$\Delta t \max \left\{ \frac{|H_1|}{\Delta x} + \frac{|H_2|}{\Delta y} + \frac{2b}{(\Delta x)^2} + \frac{2b}{(\Delta y)^2} \right\} < 1$$

19 October 2005

Michael Y. Wang

38

Signed Distance Function Re-initialization

- A **direct approach** to re-initialization is to find interface by contouring algorithm and then to compute signed distance to the newly located interface.

However, the zero level set has to be located every time the re-distance operation is called, which is very **computationally intensive**. And every time the interface is located the **error is accumulated**.

- PDE based approach** is more **preferable**, since it use only the values of the level set function and not the explicit location of the points in the domain.

19 October 2005

Michael Y. Wang

39

Signed Distance Function Reinitialization

- In **PDE based approach**, the following HJ type equation,

$$\phi_t = \text{sign}(\phi_0)(1 - |\nabla \phi|)$$

$$\phi(x,0) = \phi_0$$

is solved to steady state, which is the desired signed distance function.

- For numerical purposes it is useful to smooth the sign function, with (M. Sussman et al)

$$\text{sign}_\varepsilon(\phi_0) = \frac{\phi_0}{\sqrt{\phi_0^2 + \varepsilon^2}}$$

19 October 2005

Michael Y. Wang

40

Signed Distance Function Reinitialization

- The numeric method for reinitialization equation is,

$$\phi_j^{n+1} = \phi_j^n - \Delta t \text{Sign}_\varepsilon(\phi_j^0) G(\phi_j^n)$$

where

$$\text{sign}_\varepsilon(\phi)_j = \phi_j / \sqrt{\phi_j^2 + \varepsilon^2}$$

$$G(\phi)_j = \begin{cases} \sqrt{\max((a^+)^2, (b^-)^2) + \max((c^+)^2, (d^-)^2)} - 1, & \text{if } \phi_j^0 > 0 \\ \sqrt{\max((a^-)^2, (b^+)^2) + \max((c^-)^2, (d^+)^2)} - 1, & \text{if } \phi_j^0 < 0 \\ 0, & \text{otherwise} \end{cases}$$

$$a = D^{-x}\phi_j, \quad b = D^{+x}\phi_j$$

$$c = D^{-y}\phi_j, \quad d = D^{+y}\phi_j$$

19 October 2005

Michael Y. Wang

41

Signed Distance Function Reinitialization

- The above approach **generally works well** when ϕ_0 is neither too flat nor too steep near the interface.

However, when ϕ_0 becomes too flat, the propagating speed will be small, when ϕ_0 is very steep near the interface, this approach may change the sign of ϕ_0 , thus **moving** the interface across grid points.

- Peng et. al. improved the reinitialization equation by using a approximation to the sign function depending on the local gradient of ϕ . Their sign function is

$$\text{sign}(\phi) = \frac{\phi}{\sqrt{\phi^2 + |D\phi|^2 \Delta x^2}}$$

19 October 2005

Michael Y. Wang

42

Signed Distance Function Reinitialization

- Peng's choice of approximation to sign function solves the problem of the changing of sign of ϕ (thus moving the interface across the cell boundary) in the reinitialization step when ϕ is steep and speeds up the convergence when ϕ is flat at the interface.
- Another approach to prevent the interface straying is introduced by M. Sussman et. al., where a constraint is added to the original reinitialization equation. Their equation is

$$\phi_t = \text{sign}(\phi_0)(1 - |\nabla \phi|) + \lambda f(\phi)$$

$$\lambda = \frac{-\int_{\Omega} H'(\phi) \text{sign}(\phi_0)(1 - |\nabla \phi|)}{\int_{\Omega} H'(\phi) f(\phi)}, \quad f(\phi) = H'(\phi) |\nabla \phi|$$

19 October 2005

Michael Y. Wang

43

Signed Distance Function Reinitialization

- Reinitialization can be viewed as a form of filtering which eliminates many numerical issues while preserving the interface. For example, boundary conditions far from the interface become much less important because their effect is discarded after reinitialization.

19 October 2005

Michael Y. Wang

44

Fast Marching Method

- J. Sethian, 1996
- Special case that assumes the velocity field, F , never changes sign. That is, contour is either always expanding ($F > 0$) or always shrinking ($F < 0$)
- Convert problem to a stationary formulation on a discrete grid where the contour is guaranteed to cross each grid point at most once

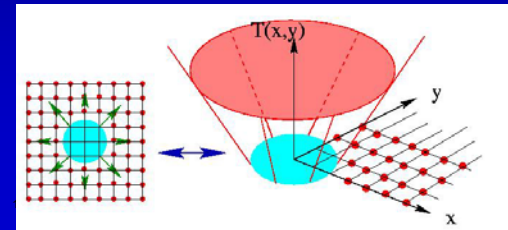
19 October 2005

Michael Y. Wang

45

Fast Marching Method

- Compute $T(\mathbf{x}, \mathbf{y})$ = time at which the contour crosses grid point (\mathbf{x}, \mathbf{y})
- At any height, t , the surface gives the set of points reached at time t



46

Fast Marching Algorithm

- Compute T using the fact that
 - **Distance = rate \times time**
 - In 1D: $1 = F \times dT/dx$
 - In 2D: $1 = F \times |\nabla T|$
- Contour at time $t =$

$$\{(x, y) \mid T(x, y) = t\}$$

19 October 2005

Michael Y. Wang

47

Fast Marching Algorithm

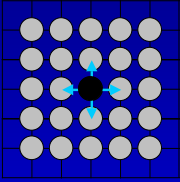
- Construct the arrival time surface $T(\mathbf{x}, \mathbf{y})$ incrementally:
 1. Build the initial contour
 2. Incrementally add on to the existing surface the part that corresponds to the contour moving with speed F (in other words, repeatedly pick a point on the fringe with minimum T value)
 3. Iterate until F goes to 0
- Builds level set surface by "scaffolding" the surface patches farther and farther away from the initial contour

19 October 2005

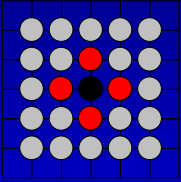
Michael Y. Wang

48

Fast Marching



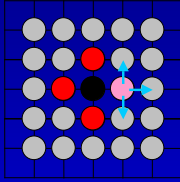
Update "downwind"
(i.e., un-visited neighbors)



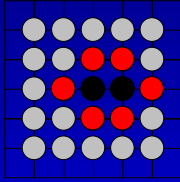
Compute new possible values

19 October 2005
Michael Y. Wang
49

Fast Marching



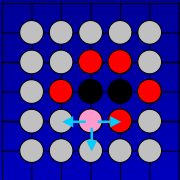
Expand point on the fringe with minimum value



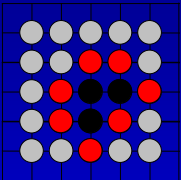
Update neighbors "downwind"

19 October 2005
Michael Y. Wang
50

Fast Marching



Expand point on the fringe with minimum value



Update neighbors "downwind"

19 October 2005
Michael Y. Wang
51


Fast Marching Visualization



19 October 2005
Michael Y. Wang
52

Example: Segmentation

- Digital Subtraction Angiogram
- F based on image gradient and contour curvature




Evolving Front, Driven by Function of Image Gradient.

19 October 2005
Michael Y. Wang
53

Example (cont.)

- Initial contour specified manually



19 October 2005
Michael Y. Wang
54

Fast Marching & Level Set for Shape Recovery

1. Fast Marching algorithm to obtain contour

$$|\nabla T|F = 1, \quad F = e^{-\alpha|\nabla G_\sigma * I(x,y)|}$$

2. Level Set algorithm

$$\frac{\partial \Phi}{\partial t} + k_f(1 - \varepsilon K)|\nabla \Phi| - \beta \nabla P \cdot \nabla \Phi = 0$$

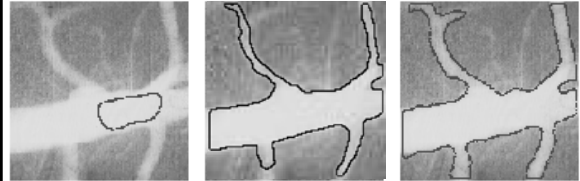
$$k_f = \frac{I}{I + |\nabla G_\sigma * I(x,y)|}$$

$$P(x,y) = -|\nabla G_\sigma * I(x,y)|$$

19 October 2005

55

Results: Vein Segmentation (continued)



Original

Fast Marching +
Level Set Tuning

Level Set only

References

- http://math.berkeley.edu/~sethian/Explanations/interface_explain.html
- S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, Springer, New York, 2003.
- J. A. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge University Press, 1999.

19 October 2005

Michael Y. Wang

57